(1)

DESIGN, FABRICATION AND TESTING OF AN ARMY BRIGADE AIR DEFENSE TERMINAL

Volume 1: Final Technical Report

Prepared by

M. Belzer, Ph. D.

Y. Cho, Ph. D.

J. Han, Ph. D.

J. Suh

S. Spriggs

Mentor Technologies, Inc.
10000N Aerospace Rd.
Lanham, MD 20706

October 7, 1992

Contract No. DAAL02-90-C-0091

Prepared for

US Army LABCOM
Adelphi Laboratory Center
2800 Powder Mill Road
Adelphi, Maryland 20783-1197

RITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

ŁEPORT SECURITY CLASSIFICATION classified	1b. RESTRICTIVE MARKINGS
;ECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT approval for public release distribution unlimited
DECLASSIFICATION/DOWNGRADING SCHEDULE	
:RFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
NAME OF PERFORMING ORGANIZATION ntor Technologies, Inc. — 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION US Army LABCOM Adelphi Laboratory Center
ADDRESS (City, State, and ZIP Code) '000 Aerospace Road, Suite N .nham, Maryland 20706	7b. ADDRESS (City, State, and ZIP Code) 2800 Powder Mill Road Adelphi, Maryland 20783-1197
NAME OF FUNDING/SPONSORING ORGANIZATION — 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER DAAL02-90-C-0091
ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

TITLE (Include Security Classification)

:sign, Fabrication, and Testing of an Army Brigade Air Defense Terminal (U)

PERSONAL AUTHOR(S)
Belzer, Y. Cho, J. Han, J. Suh, S. Spriggs

TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
.nal Technical	FROM 12Sep90 TO 11Sep92	09 Sep 92	100

SUPPLEMENTARY NOTATION

COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Multi-Sensor Multi-Target Tracking System

ABSTRACT (Continue on reverse if necessary and identify by block number)

ised on the MMAS simulation data, the functional requirements have been lentified for the HDL/CIP multi-sensor multi-target tracking system. :sign and development of software have been performed. In this effort, ıtegration with the currently existing CIP software has been the focus. ıe developed system consists of two basic modules and one supporting ıdule. The developed tracking system has also been tested by simulating :al situations. In general, for the MMAS data files with a reporting .me period of 5 seconds, the tracking performance was in acceptable ınges. However, for the data files with reporting time periods of 20 :conds or longer, miscorrelations in the data association process have en experienced.

DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified	
NAME OF RESPONSIBLE INDIVIDUAL ırry Tokarcik	22b. TELEPHONE (Include Area Code) (301) 394-3000	22c. OFFICE SYMBOL SLCHD-TA-AS

Form 1473, JUN 86 Previous editions are obsolete. SECURITY CLASSIFICATION OF THIS PAGE

Notices

Views, opinions, and/or findings contained in this report are those of the author(s) and should not be construed as an official department of the army position, policy, or decision unless so designated by other official documentation.

Table of Contents

Report Documentation Page i
Notices ii
List of Figures and Tables iii
Preface v
List of Symbols, Abbreviations and Acronyms vi
Summary ix
1. Introduction 1
1.1 Review of the CIP Software Architecture 2
2. Problem Formulation and Algorithm Development 6
2.1 Review of the MMAS Simulation Data 10
2.2 General Issues 17
2.2.1 Central Processing and Central Track File 17
2.2.2 Filtering and Data Association 18
2.2.2.1 Mechanism of the SRIF 20
2.2.2.2 Likelihood Function and the Munkres' Algorithm 24
2.3 Asynchronous Sensor Operation and Measurement Preprocessing 32
2.4 Maneuvering Targets and Multiple Model Approach 46
2.5 Low Sampling Rates and Track Update Without Measurements 57
2.6 Track Management 59
2.6.1 Track Initiation 59
2.6.2 Track Deletion 65
2.7 Other Issues 66
2.7.1 Different Types of Targets and Modeling 66
2.7.2 Multiple Filter Initialization 69
2.7.3 Adjustment of Vehicle Type 71
2.7.4 Interruption Handling 72
3. Performance Evaluation Test 73
3.1 Description of Testing Environment 73
3.2 Test Results for Each Module 74
4. Conclusion and Recommendations 96
References 99
Distributution List 100

List of Figures and Tables

Figure 1.1-1 The CIP Architecture Block Diagram

Figure 2-1 Basic Modular Architecture of the HDL

Figure 2.3-1 The Process of Discarding Old Data

Figure 2.3-2 Diagram for Time Alignment Process

Figure 2.3-3 Example of the Fusion Tree

Figure 2.3-4 Fusion Tree for the HDL

Figure 2.4-1 Example of Highly Maneuvering Target Trajectory

Figure 2.4-2 Simulated Target Trajectory

Figure 2.4-3 Change of Model Probability of ca2

Figure 2.4-4 Change of Model Probability of cv2

Figure 2.4-5 RMS Value between Simulated and Filtered State Trajectories

Figure 3.2-1 True Trajectory of Vehicle 21

Figure 3.2-2 Tracking Results with A Single Target Measurements

Figure 3.2-3 Tracking Results with Multi-Target Measurements

Figure 3.2-4 True Trajectory of Vehicle 22

Figure 3.2-5 Tracking Results with A Single Target Measurements

Figure 3.2-6 Tracking Results with Multi-Target Measurements

Table 2.1-1	Radar IEW Data Format
Table 2.1-2	Thermal Imager IEW Data Format
Table 2.1-3	Acoustic Sensor IEW Data Format
Table 2.1-4	Daysight IEW Data Format
Table 2.1-5	Available Measurements from Each Sensor Type
Table 2.3-1	Measurements with the Same Arrival Time and Detection Time
Table 2.3-2	Example of Out of Sequence Measurements
Table 2.3-3	Measurements for Standard Form of Measurement Vector
Table 2.3-4	Measurement Types Determined by the Sensors Involved in Fusion
Table 2.4-1	Model Transition Probability
Table 2.4-2	Model Initial Probability
Table 2.5-1	Example of Low Sampling Rate
Table 2.6.1-1	Measurement Data Structure
Table 2.6.1-2	Initial State Assignment
Table 2.6.1-3	Entries of Covariance Matrix with Big Number
Table 3.2-1	Input to ppm Module
Table 3.2-2	Message Buffer from ppm Module
Table 3.2-3	Track Information from Tracker Module
Table 3.2-4	Track Information from Tracker Module
Table 3.2-5	Message Buffer from ppm Module
Table 3.2-6	Track Information from Tracker Module
Table 3.2-7	Track Information from Tracker Module

Preface

This report and its companion volumes were prepared by Mentor Technologies, Inc., Lanham, Maryland, in order to satisfy documentation requirements for the effort conducted during the period September 1990 to September 1992. The algorithm design and development effort was performed under the contract DAAL02-90-C-0091, US Army LABOM, Adelphi Laboratory Center, Adelphi, Maryland. The principal investigator was Dr. M. Belzer.

The total documentation set consists of three basic volumes:

Volume 1 : Final Technical Report

Volume 2 Part A : Software Design Description

Volume 2 Part B : Source Code Listing

Volume 2 Part C : Tracking Library Functions

Volume 3 : Operator's Manual

List of Symbols, Abbreviations and Acronyms

x_k: state variable at the time k

F_k: state transition matrix

G_k: input matrix

w_k: process noise at the time k

x_0: initial state

Q_0: initial state covariance matrix

P_k: process noise covariance at the time k

y_k: measurement at the time k

H_k: observation matrix at the time k

v_k: measurement noise at the time k

$Q_v(k)$: measurement noise covariance at the time k

$R_w(k)$: process noise information matrix at the time k

$z_w(k)$: process noise information vector at the time k

$R_k(+)$: filtered state information matrix at the time k

$z_k(+)$: filtered state information vector at the time k

$R_v(k)$: measurement noise information matrix at the time k

$z_v(k)$: measurement noise information vector at the time k

$R_k(-)$: predicted state information matrix at the time k

$z_k(-)$: predicted state information vector at the time k

$\hat{x}(k \mid k-1)$: k-th predicted state estimate

$\hat{x}_k(-)$: k-th predicted state estimate

$\hat{x}(k \mid k)$: k-th filtered state estimate

$\hat{x}_k(+)$: k-th filtered state estimate

$\nu(k)$: innovation vector with k-th measurement

$S(k)$: innovation matrix associated with $\nu(k)$

(p_{ij}): model transition matrix

$\mu_j(k)$: model probability

HPWS	High Power Workstation
MMAS	Multi Mission Area Sensor
CIP	Combat Information Processor
CIM	Communication Interface Model
UAV	Unmanned Aerial Vehicle
ATA	Autonomous Target Acquisition
MHT	Multiple Hypothesis Tracker
SRIF	Square Root Information Filter
IEW	Intelligence Electronic Warfare
AC	Acoustic
DSRIF	Decentralized Square Root Information Filter
IMM	Interacting Multiple Model

Summary

The functional requirements for the HDL/CIP tracking system have been identified. The design and development of the tracking software have been performed.

Based on the tests on the MMAS simulation data file, the following are observed.

The tests show that the performance of the SRIF with the IMM is in an acceptable range. That is, even though the targets are highly maneuvering, the averaged filtered state from selected models follows closely the true trajectory without measurement noise. However, the disadvantage of this approach is in the computational load due to the use of multiple filters.

It has been observed that the choices of design parameters affect the tracking performance. For example, a change in the parameter DATA_ACCUMULATION_PERIOD leads to different sets of measurements being received by tracker module, and hence, different tracking results are sometimes expected.

An approach such as the maximum likelihood method should be considered to estimate reasonable values of parameters. The parameter estimation procedure and the tracking procedure should be performed concurrently.

The predicted state estimate is dependent on the models utilized in filtering. Also, the movement of some vehicles, especially ground vehicles, is dependent on terrain. This makes the reliability of the predicted state of ground vehicles low. An expert system which can combine kinematic information from filtering and other information, such as terrain information, should be considered.

Vehicle type information is utilized in the tracking process. The procedure employed to determine the vehicle type is very simple, and utilizes kinematic information only. The misclassification of vehicle type also leads to unexpected tracking results. The development of a system which can integrate numerical and nonnumerical information to get a more reliable vehicle type is required.

For measurements such as the MMAS data, the MHT method is more suitable than any method employing a soft decision scheme. In the implementation of the MHT, hypotheses generated must be pruned. The development of the criteria to prune the hypotheses should be considered. A hybrid system , which can integrate kinematic information and the results from the data association method employed in tracker software, and some nonnumeric information together will give more reliable data association results.

1. Introduction

With the advent of the surveillance sensor systems, more information about the battlefield is available in various forms. This information is to be put together to form an accurate and unified picture to be presented to the tactical commander. Since there is an abundant amount of information to be processed, development of a high power computing machine and supporting software is required to handle it.

Harry Diamond Laboratories (HDL) is in the process of developing a Hign Power Workstation (HPWS) as part of the Multi Mission Area Sensor (MMAS) program. The HPWS shall be designed to perform multisensor multitarget tracking of entities on the battlefield in near real time. The HPWS will be netted with various sensors on the battlefield. Sensors netted with the HPWS will send their information to the system via hard-wired or radio links. The HPWS will acquire, correlate, and process the sensor information.

The functional area analyst will access the HPWS via a tethered workstation. The user will be provided with a graphical representation of the current battlefield situation which he can query using commands entered via a mouse activated menuing system or keyboard. The system will support the analyst with knowledge-based expert systems, object-oriented order of battle and task organization databases, and a spatial database for terrain reasoning which can be accessed via an advanced man-machine interfaces. These databases will be linked via various functional area application/decision aid processes which will enable the analyst to perform his tasks more efficiently. This contract is primarily concerned with the support provided to the analyst in the areas of multisensor multitarget tracking of entities on battlefield.

This final report consists of the following. In chapter 1, the Combat Information Processor (CIP) system, the software which will support the HPWS is reviewed briefly, especially focusing the parts which are relevant to MTI's task. Chapter 2 describes some problems and the corresponding algorithms. Test results based on the MMAS simulation data and conclu-

sions are provided in chapters 3 and 4, respectively. This final report ends with the recommendations for future enhancements.

1.1. Review of the CIP Software Architecture

For system integration purpose, the architecture study about the CIP software system has been performed. The following are the descriptions of processes which are relevant to tracking task. The corresponding block diagram is provided in figure 1.1-1.

1) Message Fact Process: The sources of messages provided to the message fact process are the Communication Interface Module (CIM), the Unmanned Aerial Vehicle (UAV) Ground Data Processor, the Multi Mission Area Sensors (MMAS) Sensor Network, and the Autonomous Target Acquisition (ATA) Systems. The main purpose of this process is to collect the CIP messages and to transmit the messages to the tracker process, while archiving. The main routine starts with calling the netSTART function and continues to initialize the client interface, terrain database, and message database. Once the initialization is complete, the message fact process waits until it has received a message from the VMIM, UAV, MMAS, and ATA. When message comes in, the service command is processed. Service_cmd reads command header from command packet, and executes appropriate procedures.

2) Tracker Process: The main purpose of this routine is keeping a local copy of the acquired units list. The units list will be updated when unitcntrl sends an update command in the form of cmd packets to the tracker. The tracker compares an incoming message from the message fact process with the acquired unit list, then sends the appropriate command unitcntrl process.

3) Unit Control Process: The main purpose of this process is maintaining a master copy of database for units. The unitcntrl process sends command packets to the tracker and receive messages from the tracker, as well as maintaining unit positions and updating all attached workstations as necessary.

4) Units: Units is a task that runs on a workstation using a local copy of the master database which is maintained by the unitcntrl process. Units continuously updates the battlefield unit information and displays the result on the workstation monitor.

5) Menu Server: Menu server is an interactive program which synchronizes user actions with program responses. The Open Systems Foundation's MOTIF is used as a graphical user interface. Normally, the user is given a prompt for input of the next parameter to be supplied as part of a command sequence initiated by depressing a specific command key. When a command is acknowledged, the action informs unitcntrl to modify the data structure and update the display.

In addition to the above processes, Multiple Hypothesis Tracker (MHT) module is included for the purpose of data association. This module consists of the following three processes.

6) TRKMAN Process: The main purpose of this process is the management of tracks, which includes merging similar tracks, initiating new tracks, and deleting unnecessary tracks. Also, this process supports the processes HYP and CORR by providing new TRACKVAL and track history.

7) HYP Process: The main purpose of this process is to compute correlations and choose the best hypothesis.

8) CORR Process: The main purpose of this process is to compute likelihoods of new reports and currently existing tracks.

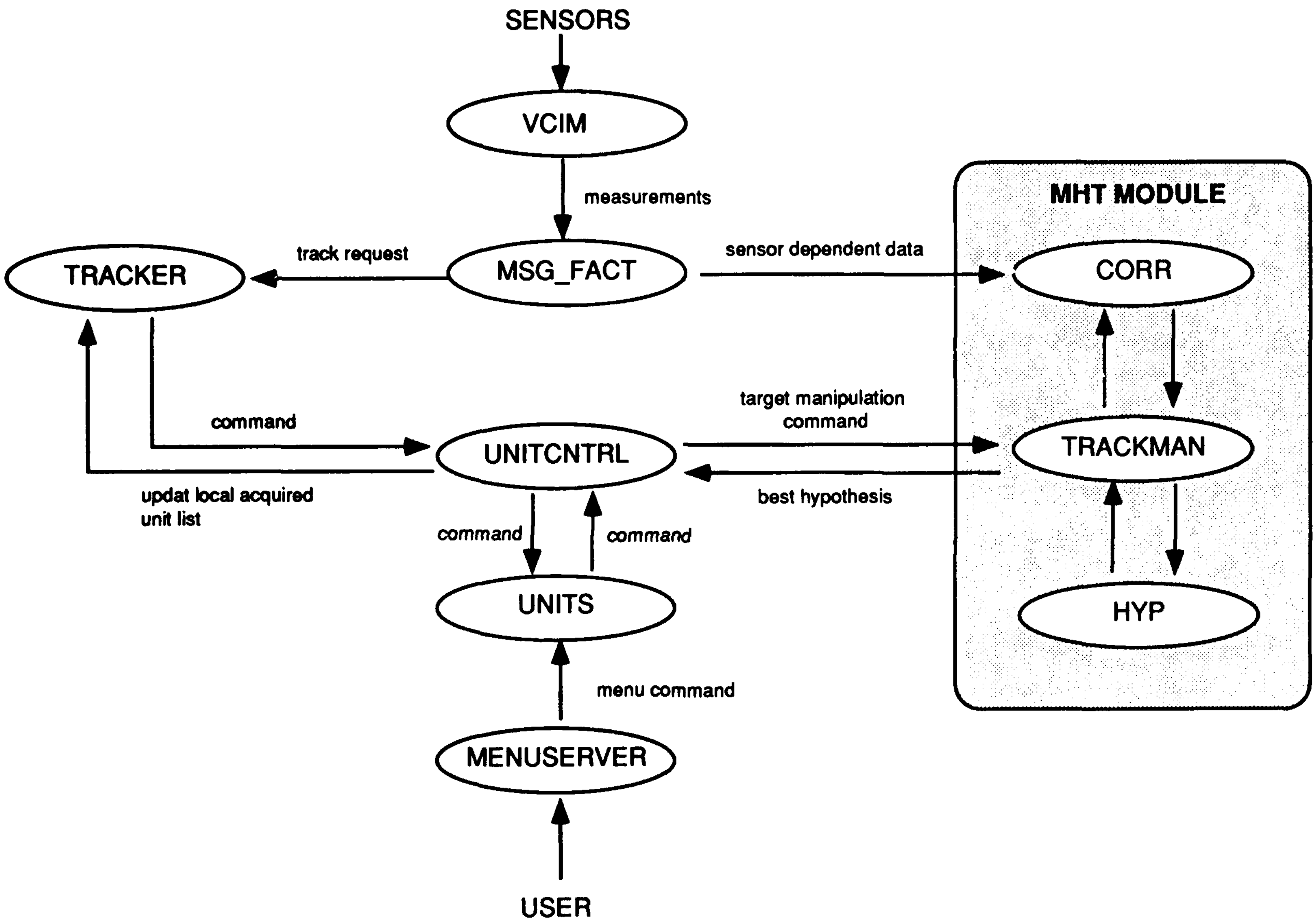


Figure 1.1-1 The CIP Architecture Block Diagram

2. Problem Formulation and Algorithm Development

The goal of the tracking software for the HDL/CIP is to generate trajectories of different types of airborne and ground vehicles based on the measurements from the multi-sensor system. The multi-sensor system provides measurements with different contexts and different dimensions.

Design of a tracking system is problem dependent. In particular, it depends on the sensors utilized and the types of targets to be tracked. The MMAS simulation data provided by HDL has been reviewed to get information about the sensors and the targets. A summary of the review is included in section 2.1.

As a result of the review, an approach to the design of the tracking system which consists of two basic modules has been employed. The first module handles the preprocessing of measurements from different types of sensors, and the other module handles the main tracking task. See figure 2-1.

As described in section 2.3, the measurement preprocessing module fuses measurements from different sensors, and provides measurement data to the tracker module in a standard format. Then, the tracker module starts the tracking task based on the fused, standard format data. By taking this approach, the design of the tracker is simplified, since it is not necessary to consider a different tracker for each different sensor type. The advantage of this design is that the tracker module is totally independent of the sensor configuration. That is, it is independent of what kind of sensors are used, and also independent of how many sensors are used. The tracker module interfaces with the multi-sensor system through the measurement preprocessing module, in that it receives the standard format measurement data that the measurement preprocessing module provides.

When more sensors are added for the enhancement of the CIP system, it may be necessary to modify the measurement preprocessing module to generate the a new standard format for the measurement data which reflects the inclusion of a new set of sensors. For the tracker module, only the interface part, i.e., the part that copies the measurements stored in the buffer to the measurement data structure for tracking, is subject to change. More details are provided in Volume 2, Part A.

The tracker module performs tasks associated with tracking different types of vehicles. The main tracking tasks, i.e., filtering and data association are discussed in section 2.2.2. Section 2.2.2.1 is devoted to the descriptions of the mechanism of the Square Root Information Filter (SRIF). The SRIF is employed since a tracking system based on the architecture of central processing and a central track file is suitable for the physical configuration of the CIP system. Discussions regarding this aspect of design of the CIP tracker are provided in section 2.2.1. The data association method for the CIP is described in section 2.2.2.2. The likelihood function and Munkres' algorithm are chosen as the main tools for this purpose.

A multiple model approach, specifically the Interacting Multiple Model (IMM) algorithm has been implemented for tracking maneuvering targets. One of the important things for the implementation of the IMM is the modeling of the movement or dynamics of the vehicles. Models based on constant velocity motion and constant acceleration motion are assumed in tracker software. If other models are added to improve tracking performance, modification of tracker module will be necessary. On the other hand, the modeling of vehicles does not affect the measurement preprocessing module. A detailed discussion is included in section 2.4.

The MMAS simulation data contains data with low sampling rates. The quality of tracking, especially the quality of filtering, is dependent on the sampling rates. An algorithm which allows tracking tasks to proceed even though measurements are not available is considered in section 2.5.

Other issues such as modeling, initialization of the filter, interrupt handling, menu system, and track file management are included in sections 2.6 and 2.7.

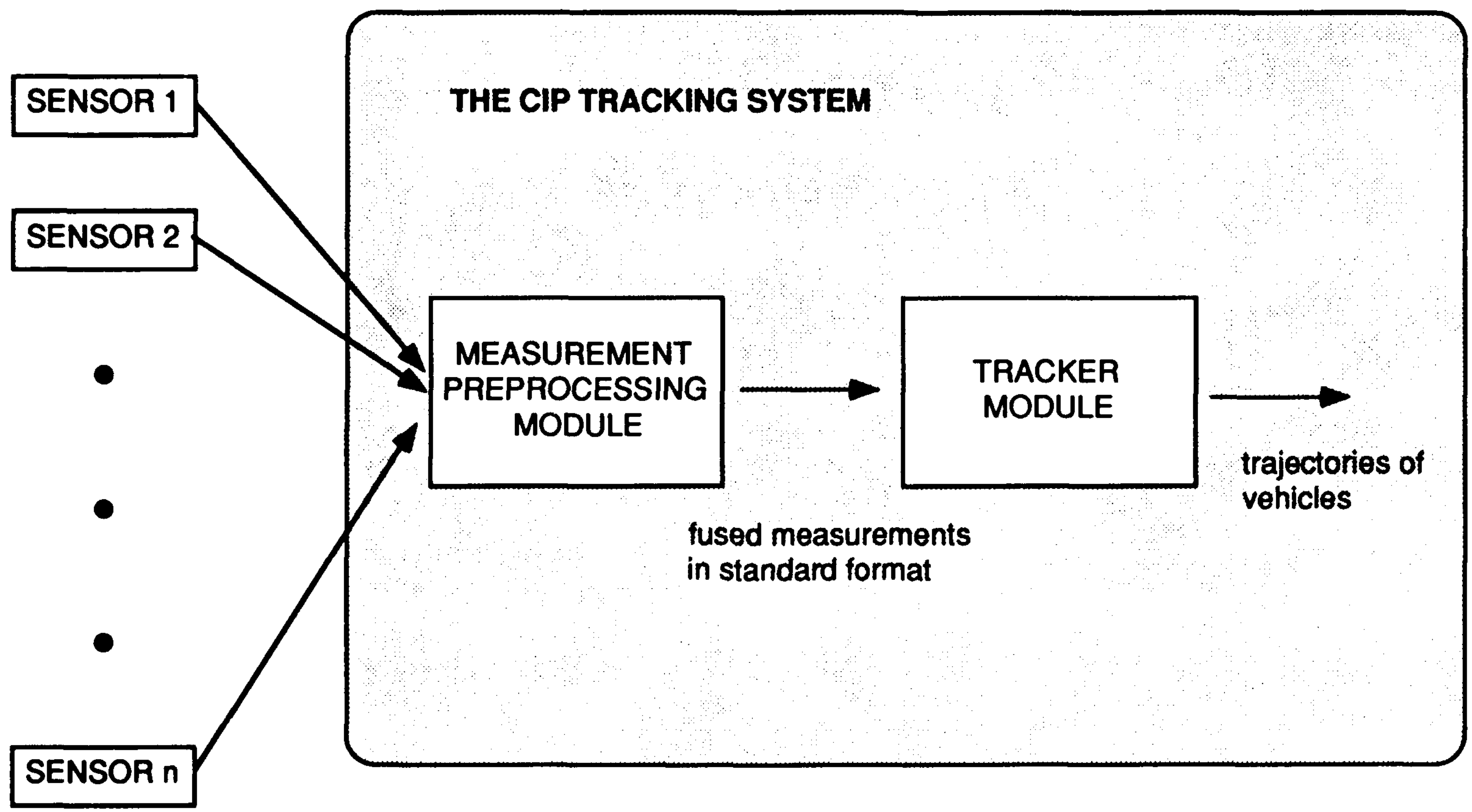


Figure 2-1 Basic Modular Architecture of the HDL/CIP Tracking System

2.1. Review of the MMAS Simulation Data

The MMAS simulation data has been provided by HDL for developing the CIP tracker software. Descriptions of the simulation data are provided in this section. The important observations made for tracking, which are relevant to asynchronous operation of multi-sensor system, are discussed in detail in section 2.3.

The simulation data consists of two categories, Intelligence and Electronic Warfare (IEW) and Acoustic (AC) data. There are 24 files of the IEW data, and 20 files of the AC data. Each file of the IEW data represents one event. For example, the file iew_d1e1 represents the simulated event 1 on the day 1. According to the MMAS file descriptions, AC data was obtained through the post processing of the IEW data.

Six different types of sensors are listed in the MMAS data file. However, as shown in table 2.1-1 through 2.1-4. there are 4 different types of data formats, i.e., radar data format, thermal imager data format, acoustic sensor data format, and daysight data format. All radars, whether they are air defense radars, fire support radars, or radars follow the radar data format.

Even though the simulation data file contains all the necessary information, only a part of information is actually available in real situations. In table 2.1-5, information which is available through real sensors is listed. The following assumptions are made with the help from HDL. The air defense radar provides the x-position, y-position, and z-position. The ground radar provides the x-position, y-position, x-velocity, and y-velocity. Measurements from the thermal imager are x-position, y-position, and vehicle type. Even though the acoustic sensor provides the target's bearing, it has not been considered in design of tracking system.

In the design of the tracking system, both of the measurement preprocessing module and the tracker module are based on the assumptions made above.

The fusion problem, i.e., how to fuse these different types of measurements mentioned above to generate a standard format of data for tracking, is considered in detail in section 2.3.

Also, the simulation data file contains information about the vehicles utilized in events. These include two types of airborne targets, A-7 and helicopter, and three types of ground vehicles, tanks, M2s, and HMMVs.

All simulations were performed as follows. The origin of the coordinate system is 32^o $18'$ N, 105^o $54'$ W. All coordinates are in meters. The sensor identifications, locations, and pointing are also included in the data file. By utilizing sensor location information with respect to the origin, a coordinate transformation of data obtained at the sensor site to the origin might be possible. However, since the data shown in the file are already coordinate transformed measurements, this aspect has not been considered in the design.

All sensors were operated in "sensor ground-truth" mode, which means that all targets within range are reported, whether or not they are detected. Detected targets have a positive detection status, while undetected targets have a zero or negative detection status.

Ground targets moved at 7.15 m/sec (except at the rough terrain location where they slowed to 0.89 m/sec) and were spaced at 50 meters apart. Helicopters flew at 50.8 m/sec at 30 m altitude, and were 10 seconds apart. A-7s flew at 250 m/sec at 30 m altitude, and were also 10 seconds apart.

sensor #	sensor number
unit #	unit number of sensor
veh #	vehicle number
veh type	vehicle type
x reported	reported x position value
error in x	error between true value of x and x reported
y reported	reported y position value
error in y	error between true value of y and y reported
z reported	reported z position value
error in z	error between true value of z and z reported
veh speed	vehicle speed
veh direction	vehicle direction
det status	detection status
sigma x	standard deviation of error in x
sigma y	standard deviation of error in y
sigma z	standard deviation of error in z
time detected	time detected
time reported	time reported to the CIP

Table 2.1-1 Radar IEW Data Format

sensor #	sensor number
unit #	unit number of sensor
veh #	vehicle number
veh type	vehicle type
x reported	reported x position value
error in x	error between true value of x and x reported
y reported	reported y position value
error in y	error between true value of y and y reported
z reported	reported z position value
error in z	error between true value of z and z reported
veh speed	vehicle speed
speed error	error in speed
veh direction	vehicle direction
direction error	error in direction
det status	detection status
error radius	error in radius
time detected	time detected
time reported	time reported to the CIP

Table 2.1-2 Thermal Imager IEW Data Format

sensor #	sensor number
unit #	unit number of sensor
veh #	vehicle number
veh type	vehicle type
det status	detection status
bearing	reported bearing measurement
bearing error	error in bearing
error diameter	error in diameter
signal to noise	signal to noise ratio
x position	x position estimated from bearing
y position	y position estimated from bearing
z position	z position estimated from bearing
veh speed	vehicle speed
veh direction	vehicle speed
time detected	time detected
time reported	time reported to the CIP

Table 2.1-3 Acoustic Sensor IEW Data Format

unit	unit number	
unit-x	x position of unit	
unit-y	y position of unit	
speed	speed of unit	
direction	direction of unit	
# of detections	number of detections	
veh spcg		
fire status	fire status	
	Obs id Unit id	observer id and unit id
	veh id	vehicle id
	veh x	x position of vehicle
	veh y	y position of vehicle
	speed	speed of vehicle
	direction	direction of vehicle
	det	detection status

Table 2.1-4 Daysight IEW Data Format

	x_posiition	y_position	z_position	x_velocity	y_velocity	veh_type
Airborne Radar	yes	yes	yes	no	no	no
Ground Radar	yes	yes	no	yes	yes	no
Thermal Imager	yes	yes	no	no	no	yes

Table 2.1-5 Available Measurements from Each Sensor Type

2.2. General Issues

The information processing architecture and the track file system are the first problems raised in multi-sensor multi-target tracking system design. The standard approach is to process the information at one place and keep the track file at the same place, i.e., a central processing and central track file system. In section 2.2.1, the discussion about this point is provided.

Another general problem in the design of tracking system is to decide the method of data association and the algorithm to integrate data association with the filter. Section 2.2.2 contains the descriptions about the mechanism of the SRIF and the method of data association. Also, procedures to combine these two main components are provided.

2.2.1. Central Processing and Central Track File

The information processing of a tracking system can be performed by either a central processing scheme or a distributed sensor-level processing scheme.

In a central processing scheme, all the information gathered through the sensors is transmitted to the central unit for processing. In this case, the central information processing unit should be equipped with high computational powers to process the data in the required time period. In most cases, the raw data observed by the sensors are sent directly to the central processing unit. This makes the communication loads between the sensors and the central unit heavy. However, the advantage of this approach is in the simplicity of the tracking algorithm.

To avoid the heavy computational load on the central processing unit and to reduce communication costs, a distributed sensor-level processing scheme is suggested. In this approach, raw measurement data are preprocessed up to certain level at the sensor site before they are sent to the central processing unit. The main disadvantage of this scheme is that the tracking algorithm becomes complicated.

In reviewing the MMAS simulation data, it has been observed that all measurements obtained by sensors are sent directly to the CIP for further processing. This suggests that the central processing scheme is a more natural approach for this application than the sensor-level distributed processing.

On the other hand, since the MMAS simulation data file shows that no sensor has any capability other than sending information to the CIP, it is not necessary to keep track files at all the sensor sites. Hence, the central track file system is adopted for the CIP system.

According to the information provided by HDL, the CIP system will be mounted on a small truck. If several trucks with CIP systems are connected to allow inter-communications through local or wide area networks, then a distributed network of tracking system is formed. In this case, the software developed herein can be used as it is. However, the track files at all the CIP sites form a networked file system, and the track management algorithm should be modified to incorporate any track information sent by other CIP systems.

2.2.2. Filtering and Data Association

The filtering process is a key component of the tracking process. It may be possible to track a target simply by observing consecutive measurement data, but if measurements are missed because of a faulty sensor, a track might be lost. Additionally, since measurement data can be corrupted by noise, including electronic noise inherent to the sensor, tracking

based only on measurement data may not perform well enough when an accurate position of the target is required. The case where the tactical commanders must make decisions based on the current battlefield situation is illustrative of the level of accuracy that measurement data alone cannot provide.

The SRIF is utilized for the filtering process. The reason why the SRIF was chosen, instead of the Decentralized Square Root Information Filter (DSRIF), is that the SRIF is more suitable to central processing than the DSRIF. In the following section, 2.2.2.1, the mechanism of the SRIF is described.

Two issues are involved in data association. The first one is measurement-to-measurement association, and the second is measurement-to-track association. The first one will be discussed in detail in section 2.3, associated with measurement fusion, and only the second is considered here.

When the CIP receives data about several targets from several sensors, it is necessary to decide which data are associated with which currently existing tracks. This is the measurement-to-track association problem.

There are several approaches to this problem, and these approaches are usually categorized as hard decision schemes or soft decision schemes. In a hard decision scheme, the final decision of association is made at the time when the measurements are received and the currently existing tracks are updated. The assignment matrix, the nearest neighborhood method, the branching algorithm, and the likelihood function are the well known conventional approaches or tools categorized as hard decision schemes.

In a soft decision scheme, the final decision is delayed to allow collection of more information. The multiple hypothesis tracking method is the well known approach belonging to

this category. The multiple hypothesis tracking method has been chosen as a data association method for the CIP by HDL. However, to provide more flexibility to the CIP system, our approach is a hard decision scheme based on a likelihood function and Munkres' algorithm. These are discussed in section 2.2.2.2 in detail.

2.2.2.1. Mechanism of the SRIF

The derivation of the SRIF is not considered in this section, since it can be found in several references, including [1]. Instead, a description of the mechanism of one cycle of a track update based on the SRIF is given below.

First, assume that a dynamic model for a target is given by the linear form

$$x_{k+1} = F_k x_k + G_k w_k. \tag{2.2.2.1-1}$$

Here, x_k represents a state vector at the time k, which consists of kinematic information such as position, velocity, and acceleration.

To initialize the SRIF, initial state x_0 and its associated statistics are required a priori. It is assumed that $x_0 \sim N(\overline{x_0}, Q_0)$. Here, $\overline{x_0}$ and Q_0 represent the corresponding mean vector and covariance, respectively.

Next, w_k represents process noise vector at the time k, and it is also assumed that $w_k \sim N(\overline{w_k}, P_k)$. $\overline{w_k}$ and P_k are the process noise mean vector and its covariance, respectively.

F_k and G_k are determined by the model of target dynamics. Section 2.7.1 contains a more detailed discussion about the form of F_k and G_k.

Second, assume a linear measurement model given by

$$y_k = H_k x_k + v_k \tag{2.2.2.1-2}$$

Here, y_k is the k-th measurement vector associated with a certain type of sensor. v_k represents measurement noise vector. Like process noise, it is assumed that $v_k \sim N(0, Q_v(k))$. Here Q_v represents measurement noise error covariance.

H_k is the observation matrix, and depends on what kind of measurements are available through a specific sensor.

To implement the SRIF, a factorization of the covariance matrices into a product of information matrices must be done a priori, and the corresponding information vectors should be obtained. From process noise covariance, we get

$$P_k = R_w^{-1}(k) R_w^{-T}(k) \quad \text{and} \quad z_w(k) = R_w(k)\overline{w_k}.$$

Similarly,

$$Q_0 = R_0(+)^{-1} R_0(+)^{-T} \quad \text{and} \quad z_0(+) = R_0(+)\overline{x_0} \quad ,$$

$$Q_v(k) = R_v(k)^{-1} R_v(k)^{-T} \quad \text{and} \quad z_v(k) = R_v(k) \cdot 0 = 0 \quad ,$$

from initial state statistics and measurement noise statistics, respectively. Here, R_w and z_w are called the process noise information matrix and the process noise information vector, respectively. $R_0(+)$ and $z_0(+)$ are called the 0-th filtered state information matrix and information vector, respectively. Also, R_v and z_v are called the measurement noise information matrix and information vector, respectively. The superscript T represents the transpose of the matrix.

Then one cycle of the SRIF mechanism from step k to $k+1$ starts with time update which is given by

$$T\begin{bmatrix} R_w(k) & 0 & z_w(k) \\ -R_k(+)F_k^{-1}G_k & R_k(+)F_k^{-1} & Z_k(+) \end{bmatrix}$$

$$= \begin{bmatrix} R_w^*(k+1) & R_{wx}^*(k+1) & z_w^*(k+1) \\ 0 & R_{k+1}(-) & z_{k+1}(-) \end{bmatrix} \qquad (2.2.2.1\text{-}3)$$

Here, $R_w^*(k+1)$, $R_{wx}^*(k+1)$, and $z_w^*(k+1)$ are called the smoothing coefficients, which are not used in filtering. $R_{k+1}(-)$ and $z_{k+1}(-)$ are $k+1$ step of predicted state information matrix and information vector, respectively. T is an orthogonal transformation from Householder's method used to make the matrix on the right side of the equation (2.2.2.1-3) upper triangular.

The results from the time update step, in particular the predicted state information matrix $R_{k+1}(-)$ and information vector $z_{k+1}(-)$ are utilized in measurement update step. During measurement updating, new measurements are incorporated to generate the filtered state information matrix and information vector as follows.

$$T\begin{bmatrix} R_{k+1}(-) & z_{k+1}(-) \\ R_v H_{k+1} & R_v y_{k+1} \end{bmatrix} = \begin{bmatrix} R_{k+1}(+) & z_{k+1}(+) \\ 0 & e_{k+1} \end{bmatrix} \qquad (2.2.2.1\text{-}4)$$

Here, e_{k+1} is called the measurement error, and its norm square represents the normalized form of the norm of the innovation vector. This is a very important result and this has been utilized for the evaluation of likelihood function in terms of the SRIF variables only. See section 2.2.2.2 for more details.

Since measurement noise has a covariance which is not in general an identity matrix, a whitening process, i.e., multiplication of measurement information matrix by H_{k+1} and y_{k+1} has been performed.

The filtered state information matrix, $R_{k+1}(+)$, and information vector $z_{k+1}(+)$ are fed into equation (2.2.2.1-3) to repeat the cycle.

Note that the information matrix and vector obtained from the statistics of the initial state are used as the initial filtered state information matrix and information vector to initiate the time update step. However, they can also be used as the initial predicted state information matrix and information vector. In this case, the filtering cycle should start with the measurement update step. Both approaches give the same filtering result.

As mentioned before, the implementation of the SRIF is also based on the observations made about the MMAS simulation data file. This enables us to use the SRIF, not the extended form of the SRIF. In the general situation, the dynamic model or measurement model or both do not take linear forms. To achieve a better filtering performance, more accurate modeling is required. This usually results in more complicated nonlinear models. In this case, the extended form of the SRIF should be utilized as follows.

For a given nonlinear system,

$$x_{k+1} = f(x_k) + Gw_k \ , \quad \text{and}$$

$$y_k = h(x_k) + v_k$$

the extended form of the SRIF requires linearization procedures to get

$$y_k = H_k x_k + v_k + z_k$$

where

$$z_k = h(x)|_{x = x_k(-)} - H_k x_k(-) \quad , \quad H_k = \frac{\partial h(x)}{\partial x}|_{x = x_k(-)}$$

and

$$x_{k+1} = F_k x_k + B_k w_k + g_k$$

where

$$g_k = f(x)|_{x = x_k(+)} - F_k x_k(+) \quad , \quad F_k = \frac{\partial f(x)}{\partial x}|_{x = x_k(+)} .$$

Then the time update and measurement update are given as follows.

Time Update:

$$T \begin{bmatrix} R_w(k) & 0 & z_w(k) \\ -R_k(+)F_k^{-1}G & R_k(+)F_k^{-1} & z_k(+) + R_k(+)F_k^{-1}g_k \end{bmatrix} = \begin{bmatrix} R_w^*(k+1) & R_{wx}^*(k+1) & z_w^*(k+1) \\ 0 & R_{k+1}(-) & z_{k+1}(-) \end{bmatrix}$$

Measurement Update:

$$T \begin{bmatrix} R_{k+1}(-) & z_{k+1}(-) \\ R_v(k+1)H_{k+1} & R_v(k+1)(y_{k+1} - z_{k+1}) \end{bmatrix} = \begin{bmatrix} R_{k+1}(+) & z_{k+1}(+) \\ 0 & e_{k+1} \end{bmatrix}$$

2.2.2.2. Likelihood Function and Munkres' Algorithm

In the parameter estimation theory, the likelihood function is one of the several methods which have been used frequently. For a given sequence of measurements

$$y(1), y(2), ..., y(k) \tag{2.2.2.2-1}$$

up to k, the likelihood function of the filter is defined by the joint probability density function $P[y(1), \cdots, y(k)]$ [2]. Then, by simple calculation utilizing the Bayesian formula, we get

$$P[y(1), \cdots, y(k)] = \prod_{i=1}^{k} P[y(i) \mid Y^{i-1}] \qquad (2.2.2.2\text{-}2)$$

where $Y^{i-1} = \left\{ y(j)\,,\ j = 1, \ldots, i-1 \right\}$.

Under the assumption that all conditional probability density functions in (2.2.2.2-2) are Gaussian, we have

$$P[y(i) \mid Y^{i-1}] \sim N(\nu(i);\, 0;\, S(i)) \qquad (2.2.2.2\text{-}3)$$

where $\nu(i)$, and $S(i)$ are the innovation vector and its covariance, respectively. Using (2.2.2.2-3) in (2.2.2.2-2), the likelihood function of the filter takes the form

$$\left[\prod_{i=1}^{k} \det(2\pi S(i))^{-1/2} \right] \exp\left[-1/2 \sum_{i=1}^{k} \nu(i)^{t} S^{-1}(i) \nu(i) \right] \qquad (2.2.2.2\text{-}4)$$

However, it is equation (2.2.2.2-3), not (2.2.2.2-4), that is useful in data association, especially measurement-to-track association. Measurement-to-track association is the process which correlates the existing tracks with new measurements. Suppose that a certain track has been formed up to time i, and that a set of measurements, $y(1), \cdots, y(i-1)$, has been utilized to update the track. Suppose that the i-th measurement data has become available. To update tracks based on the new measurement, a criterion for measurement-to-track association is required, and equation (2.2.2.2-3) represents the conditional probability of the i-th measurement given measurements up to $i-1$. Then, it is plausible to choose the measurement which gives the highest conditional probability.

Sometimes it is useful to use a modified form of (2.2.2.2-3), especially the log-likelinood form. Let's take -2 log on both sides of (2.2.2.2-3). Then we have, after disregarding the constant term $(2\pi)^{-1/2}$,

$$\det S(i) + \nu(i)^t S^{-1}(i)\nu(i) \tag{2.2.2.2-5}$$

Since $-2 \log(\cdot)$ is a monotonically decreasing function, the association of the highest conditional probability corresponds to the one which gives the smallest value in (2.2.2.2-5)

As can be seen in (2.2.2.2-3) and (2.2.2.2-5), by evaluating the two terms

$$\det S(i) \quad \text{and} \quad \nu(i)^t S^{-1}(i)\nu(i) \tag{2.2.2.2-6}$$

the values of likelihood function and log-likelihood function can be obtained. There are two ways to compute these two terms. The first approach is based on the direct computations, i.e., to compute innovation vector $\nu(i)$ and its covariance $S(i)$ using the equations

$$\nu(i) = y_i - H\hat{x}(i \mid i-1), \quad \text{and}$$

$$S(i) = HP(i \mid i-1)H^t + Q_\nu(i), \tag{2.2.2.2-7}$$

respectively. Then, by direct substitution of (2.2.2.2-7) into (2.2.2.2-6), the values of (2.2.2.2-3) and (2.2.2.2-5) are obtained. Another approach is based on the output from the SRIF. In the paper [3], it has been shown that the two terms in (2.2.2.2-6) can be evaluated in terms of the SRIF variables only. That is,

$$\det S(i) = \left[\frac{\det R_i(+)}{\det R_i(-)\det R_\nu} \right]^2 \quad \text{and}$$

$$\nu(i)^t S^{-1}(i)\nu(i) = \| e_i \|^2. \tag{2.2.2.2-8}$$

The advantage of the first approach is that only the time update step is required to get innovation vector and its covariance. However, as shown in (2.2.2.2-6), it requires computing the determinant and inverse of the covariance matrix, which is computationally expensive. The advantage of the second method is that it is relatively simple, as described in (2.2.2.2-8). The disadvantage of this approach is that measurement update step of the SRIF should be

completed a priori.

Measurement-to-track association can be viewed as an assignment problem. That is, the assignment of existing tracks to newly obtained measurements. For an assignment problem, there are several well known algorithms including Munkres' algorithm [4].

Munkres' algorithm accepts as input a matrix of variable size, called an assignment matrix, where one dimension is indexed by the set of tracks and the other is indexed by the set of measurements. The entries of the matrix must be non-negative, and must have the property that higher values denote a worse match between the measurement and the track. In other words, (i, j) entry of an assignment matrix represents the cost which should be paid by associating i-th track with j-th measurement. The statistical distance $\nu(i)' S^{-1}(i)\nu(i)$ has the required properties, as does the log-likelihood value in (2.2.2.2-5).

Once the assignment matrix has been set up, Munkres' algorithm goes through an iterative procedure to find a pairing of tracks and measurements, such that the smaller set is exhausted and the total cost is minimized. Here, the total cost is defined as the sum of the corresponding individual costs for one possible set of assignments. In other words, Munkres' algorithm chooses the set of assignments whose total cost is minimum. To illustrate the idea, let's consider the following assignment matrix.

	measurement 1	measurement 2
track 1	0.6	0.2
track 2	0.7	0.2

Both track 1 and track 2 are competing for measurement 2, since the association costs are smaller then the association costs with measurement 1. In this case, there are two possi-

ble sets of assignments. That is, (track 1 - measurement 1, track 2 - measurement 2), and (track 1 - measurement 2, track 2 - measurement 1). The corresponding total costs are 0.8 and 0.9, respectively. Hence, the first set of assignments is decided as an optimal one.

In some cases, the assignment matrix is not the square matrix. Sometimes the number of tracks is larger than the number of measurements. Then, there are some tracks which are not associated with any measurements, and track update without measurements follows. In the converse case, i.e., when the number of measurements is larger than the number of currently existing tracks, the measurements which are not associated with any tracks are regarded as new measurements, and the new track initiation process will proceed. Even when the number of tracks and measurements are equal (the square matrix case), an individual association whose cost is too high will then be disassociated and treated as a track without a measurement and a measurement without a previous track.

A slight variation of this that is also implemented is to apply a gating test to the each possible association before Munkres' algorithm is applied. An association that failed the gating test would have a large value assigned instead of its original cost. Processing would then be as before.

For a set of new measurements $y_1(i), \cdots ,y_n(i)$ at the time i and m currently existing targets, the following steps summarize the data association procedure.

step 1:

Choose a target and compute $H\hat{x}(i \mid i-1)$. Here $\hat{x}(i \mid i-1)$ is the predicted state of the target at the i-th time instant and H is the observation matrix. Hence, $H\hat{x}(i \mid i-1)$ represents the predicted measurement of the target at the time i.

step 2:

Compute the innovation $\nu_k(i) = y_k(i) - H\hat{x}(i \mid i-1)$. Here $y_k(i)$ represents the k-th measurement at the time i.

step 3:

Compute the corresponding innovation covariance $S_k(i)$ using

$$S_k(i) = HQ(i \mid i-1)H^t + Q_v^k(i).$$

Here $Q(i \mid i-1)$ is the predicted state error covariance and $Q_v^k(i)$ is the measurement noise covariance associated with $y_k(i)$.

step 4:

Compute the log-likelihood value $l_k(i)$ using

$$l_k(i) = \det S_k(i) + \nu_k(i)^t S_k^{-1}(i)\nu_k(i) .$$

step 5:

Repeat step 1 through step 4 for each pair of target and measurement to form an $m \times n$ matrix L whose (p, q) entry is the log-likelihood value between the p-th target and q-th measurement.

step 6:

Search for the maximum value of all entries of L.

step 7:

Perform gating test and adjust the matrix. For each pair of (p, q), if it passes gating test, then keep the original log-likelihood value. Otherwise, switch the corresponding log-likelihood value to the maximum value obtained in step 6.

step 8:

Apply Munkres' algorithm. The Munkres' algorithm provides an optimal assignments between target group and measurement group.

step 9:

Perform disassociation process. For each associated pair obtained from the Munkres' algorithm, check gating result. If it has already passed gating test, the association survives. Otherwise, disassociate the pair since it has failed gating test.

From step 9, the final measurement-to-track association is obtained. Note that one target is associated with exactly one measurement. For the target which is not associated with any measurement, the process of measurement update without measurement is performed, and the number which counts this process is decreased by 1. See section 2.6.2 for more details. For the measurement which is not associated with any currently existing tracks is regarded as a new track initiator and track initiation process is invoked. See section 2.6.1 for more details.

Threshold value was usually decided by referring to the Chi-square distribution table and the value $\nu_k(i)^t S_k^{-1}(i)\nu_k(i)$. Since the value $\nu_k(i)^t S_k^{-1}(i)\nu_k(i)$ depends on dynamic models employed for filtering and the movements of some vehicles, especially ground vehicles, are constrained by terrain, the above conventional approach to decide gating threshold may not suitable for the HDL/CIP.

The approach adopted here is as follows. For each track, a gate circle centered at the $\hat{x}(i-1 \mid i-1)$, the $(i-1)$-th filtered state estimate, is formed. The radius of the gate circle is determined by

$$d_{max} = MAX\_VEH\_SPEED \cdot \Delta T(i)$$

Here $\Delta T(i)$ is the time interval between $(i-1)$-th step and i-th step, and MAX_VEH_SPEED is the maximum speed of the vehicle. Hence d_{max} represents the maximum distance the vehicle can move for the time period $\Delta T(i)$. However, MAX_VEH_SPEED depends on the vehicle type. In section 2.7.3, a method to decide vehicle type is discussed.

Finally, integration of the log-likelihood function evaluation and Munkres' algorithm with the SRIF in the multiple model environment is discussed in section 2.4.

2.3. Asynchronous Sensor Operation and Measurement Preprocessing

Sensors scan their scanning volumes periodically with fixed scanning time intervals. The scanning time intervals of sensors are different from each other. Each sensor detects targets in its scanning volume and transmits measurement data and detection times of all detected targets. There are transmission delays, which are regarded as random. The delay is usually dependent on several physical factors such as the distance between the sensor and the CIP system, the data transmission rate of each sensor, data size, and etc. Hence, the data which the CIP receives from all sensors usually do not represent the measurement detected at the same time instant. This is why we think that sensors are operated asynchronously. The following observations from the IEW data files are due to asynchronousity.

First, let us consider the set of data which arrive at the CIP at the same time (i.e., with the same time_received tags). Then, they usually contain information about a target detected at different times. In this case, after sorting the data sequentially according to the time detected, it is possible to process one measurement at a time. The interesting case is as follows.

1) Measurements with the same arrival and detection time: In table 2.3-1 obtained from the IEW data file iew_d1e3, it is observed that measurements for a specific target 21 from sensors 4001 and 4002 are made at the same time.

sensor id	veh type	time det	time rep
4002	21	2345.0	2342.5
4001	21	2345.0	2342.5

Table 2.3-1 Measurements with the Same Arrival Time and Detection Time

Another interesting case is when out of sequence measurements are observed. Let us consider two consecutive sets of measurement data which arrive at the CIP in sequential order.

2) Out of sequence measurements: In table 2.3-2 obtained from the IEW data file iew_d2e2, sensor 2001 detected vehicle #22 prior to sensor 4001's detection of vehicle #22, but sensor 4001 reported its detection earlier.

sensor id	veh type	time det	time rep
2001	22	43.0	43.0

sensor id	veh type	time det	time rep
4002	22	45.0	42.5
4001	22	45.0	42.5

Table 2.3-2 Example of Out of Sequence Measurements

To resolve the above problems, a measurement preprocessing which consists of three stages is implemented. The first stage is to receive, sort, and store measurement data. The second stage is measurement-to-measurement association procedure, and the third stage is measurement fusion procedure.

As described in section 2.1, three types of sensors, airborne radar, ground radar, and thermal imager are considered as sources of measurement data. Also, available measurements from each sensor are listed in table 2.1-5. The measurement preprocessing module will

receive measurements through the msg_fact module of the HDL/CIP. However, at this moment, the measurement preprocessing module is designed to work independently. That is, it receives one measurement from one sensor at a time without any communication with msg_fact.

RECEIVE/SORT/STORE: Once one measurement arrives at measurement preprocessing module, the time when measurement was obtained (i.e., time_det in data structure) is compared with the time_limit. The time_limit is determined by subtracting DELAY (user defined parameter) from the current time. If measurement is old enough, i.e., time_det is less than the time_limit, it is discarded since old data degrades the real-time performance of the CIP. In figure 2.3-1, any measurements falling in the shaded region, which represents the time region before the time_limit, is discarded.

For those measurements which pass the discarding procedure, sensor identification (sensor ID) is verified. That is, measurement is sorted according to its source, sensor ID.

The concept of linked list is employed to store the data. Here, linked list consists of a sequence of measurements from a single specific sensor and listed in the order of detected time from the top of the list. Storing process starts with searching for the corresponding list. If corresponding list is found, time_det of the new measurement is compared with the detection times of the measurements on the list, and the new measurement is stored at the right position of the list according to its time_det. Otherwise, it creates its own list representing its own sensor ID, and is stored at the top of the list.

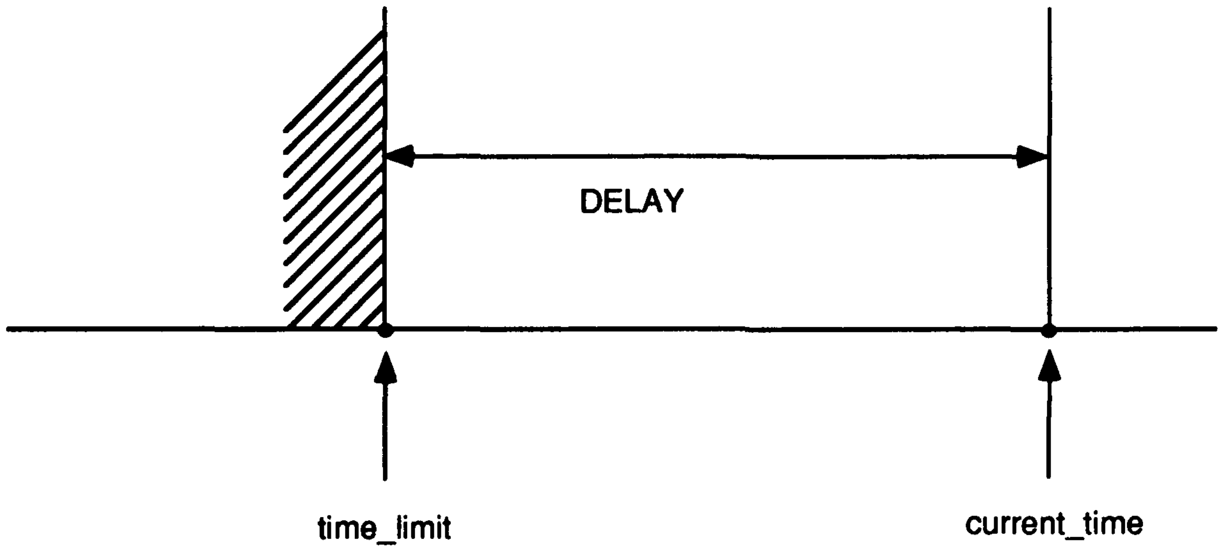


Figure 2.3-1 Time Region to Discard Old Measurements

As seen in table 2.1-5, each sensor provides different contexts of measurements. To implement fusion process readily later, 5-dimensional vector consisting of x, y, z, $\dot{x}$, and $\dot{y}$ are considered as a standard form of measurement vector.

When the measurement stored, sensor id is checked first. According to the sensor id, available data from the sensor are copied into the appropriate positions of the 5-dimensional standard form of measurement vector. Also, the corresponding components of covariance are copied. After that the remaining components of the standard measurement vector are set to zero with big number of covariance. The table 2.3-3 illustrates what measurements are copied directly into the standard measurement vector and what components are set to zero according th the sensor types.

MEASUREMENT-TO-MEASUREMENT ASSOCIATION: Data are collected for a certain period of time, defined as the DATA_ACCUMULATION_INTERVAL. Then, measurement preprocessing module pauses in receiving data and processes a transmission, which starts with measurement-to-measurement association. The measurement-to-measurement association process determines whether the measurement reports from different local sensors to the CIP have common sources of measurements.

To start measurement-to-measurement association, all measurements are represented in terms of one fixed coordinate system (e.g., a coordinate system residing at the CIP) through appropriate coordinate transformations. In the MMAS simulation data, it is assumed that necessary coordinate transformations have already been performed. This process is called spatial alignment.

Once spatial alignment is finished, time alignment should be performed. Figure 2.3-2 illustrates time alignment process. All the measurements inside the shaded area, i.e., measurements lying between the previous time_limit and the current time_limit are considered detected at the same time.

	Measurements from Sensors	Measurements Set to Zero
Air Radar	x-pos, y-pos, z-pos	x-vel, y-vel
Ground Radar	x-pos, y-pos, x-vel, y-vel	z-pos
Thermal Imager	x-pos, y-pos	z-pos, x-vel, y-vel

Table 2.3-3 Measurements for Standard Form of Measurement Vector

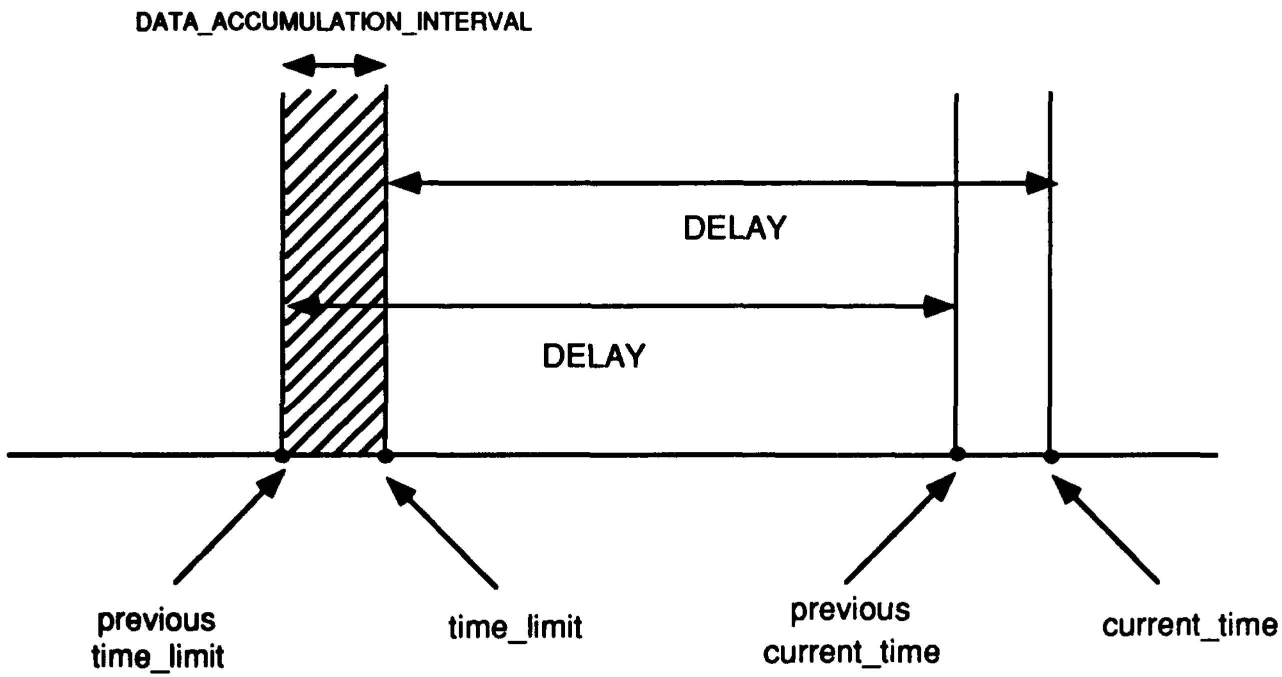


Figure 2.3-2 Diagram for Time Alignment Process

After the spatial and time alignments, measurement-to-measurement association starts. The association process begins with two data sets from any two sensors. The first measurement from the first sensor is compared successively with the measurement from the second sensor until either they are exhausted or a match is found. A match is declared when a statistical distance measure of distance falls below a threshold value. As in table 2.1-5, the measurements provided by the sensors under consideration take different forms. When types of sensors located at local sites are different, measurement characteristics might be also different. In this case the spatial association must be made in the common dimensions of measurements. The common measurements from all three sensors are x-position and y-position. Distance measure which is utilized is statistical distance which is defined as follows.

For a given pair of measurements y_1, y_2, define

$$d^2 = (y_1 - y_2)^T S^{-1} (y_1 - y_2) \tag{2.3-1}$$

where S is the covariance matrix for $y_1 - y_2$. Let $y_{1,tr}$ and $y_{2,tr}$ be true values of y_1 and y_2, respectively. Define

$$y_{1,er} = y_1 - y_{1,tr} \ ,$$

$$y_{2,er} = y_2 - y_{2,tr} \ .$$

Then

$$(y_1 - y_2) - (y_{1,tr} - y_{2,tr}) = y_{1,er} - y_{2,er} \ .$$

Under the assumption that the measurements have the same measurement source, we have $y_{1,tr} = y_{2,tr}$, and then $y_1 - y_2 = y_{1,er} - y_{2,er}$. Then the covariance becomes

$$E[(y_1 - y_2)(y_1 - y_2)^T] = E[(y_{1,er} - y_{2,er})(y_{1,er} - y_{2,er})^T]$$
$$= E[y_{1,er} {y_{1,er}}^T] + E[y_{2,er} {y_{2,er}}^T]$$
$$= R_1 + R_2 \qquad (2.3\text{-}2)$$

where R_1 and R_2 are measurement error covariances associated with y_1 and y_2, respectively.

Substituting (2.3-2) into (2.3-1) yields

$$d^2 = (y_1 - y_2)^T (R_1 + R_2)^{-1} (y_1 - y_2) \qquad (2.3\text{-}3)$$

Next, a threshold value should be determined according to the characteristics of the given problem. For example, by assuming the error difference term $y_{1,er} - y_{2,er}$ forms a Gaussian distribution, we can see easily that the metric d^2 defined in (2.3-1) follows a Chi-square (χ^2) distribution. Then Chi-square (χ^2) test can be applied by utilizing a threshold from the Chi-square (χ^2) table.

MEASUREMENT FUSION: Measurement fusion can be achieved as follows. Measurement data y_1, y_2, ..., y_n are assumed to be from one measurement source. Assume also that they are independent measurements. Let R_i be a covariance matrix associated with y_i. Then a composite measurement covariance is defined by

$$R^{-1} = \sum_{i=1}^{n} R_i^{-1} \qquad (2.3\text{-}4)$$

Then the fused measurement vector is given by

$$y = R \, [\sum_{i=1}^{n} R_i^{-1} y_i] \qquad (2.3\text{-}5)$$

As an example, let $A_1(t)$, $A_2(t)$ be the scalar measurements from sensor 1 and sensor 2 at the time instant t, respectively.

Under the assumption that each measurement has mean zero and covariances σ^2 and δ^2, respectively, then, from (2.3-4) and (2.3-5), the fused covariance R is given by

$$R = \frac{\sigma^2 \delta^2}{\sigma^2 + \delta^2}$$

and the fused measurement A is given by

$$A = \frac{\delta^2 A_1 + \sigma^2 A_2}{\sigma^2 + \delta^2}$$

Note that the fusion procedure described in (2.3-4) and (2.3-5) can be implemented in two ways. As depicted in figure 2.3-3, fusion is achieved at one time utilizing all measurements, following the equations in (2.3-4) and (2.3-5) directly. Another method is, as depicted in figure 2.3-4, fusion is started with measurements from two sensors. Then, fused result and new measurement from the third sensor are fused again. This procedure is repeated until all measurements are exhausted. The second approach has been adopted for the CIP tracker software.

For the measurement update of filtering process, measurement types are identified according to the sensors involved in fusion step. See table 2.3-4.

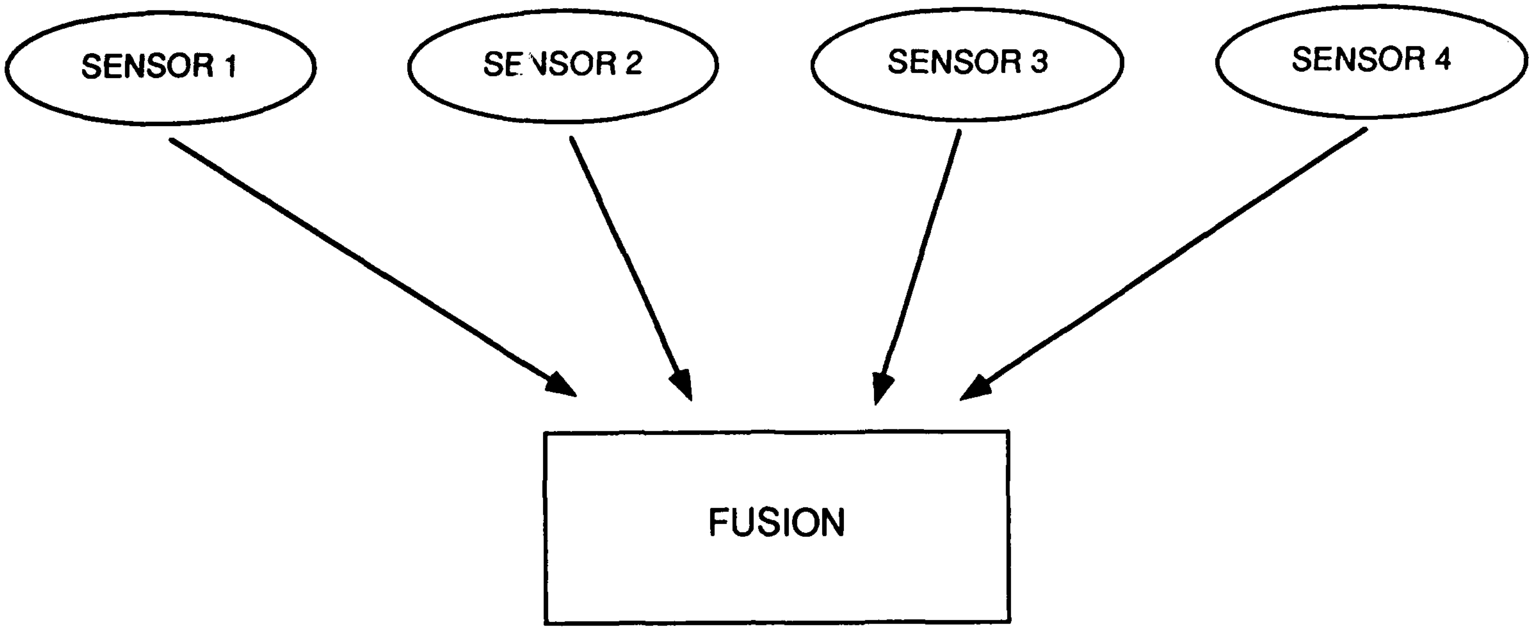


Figure 2.3-3   Example of the Fusion Tree

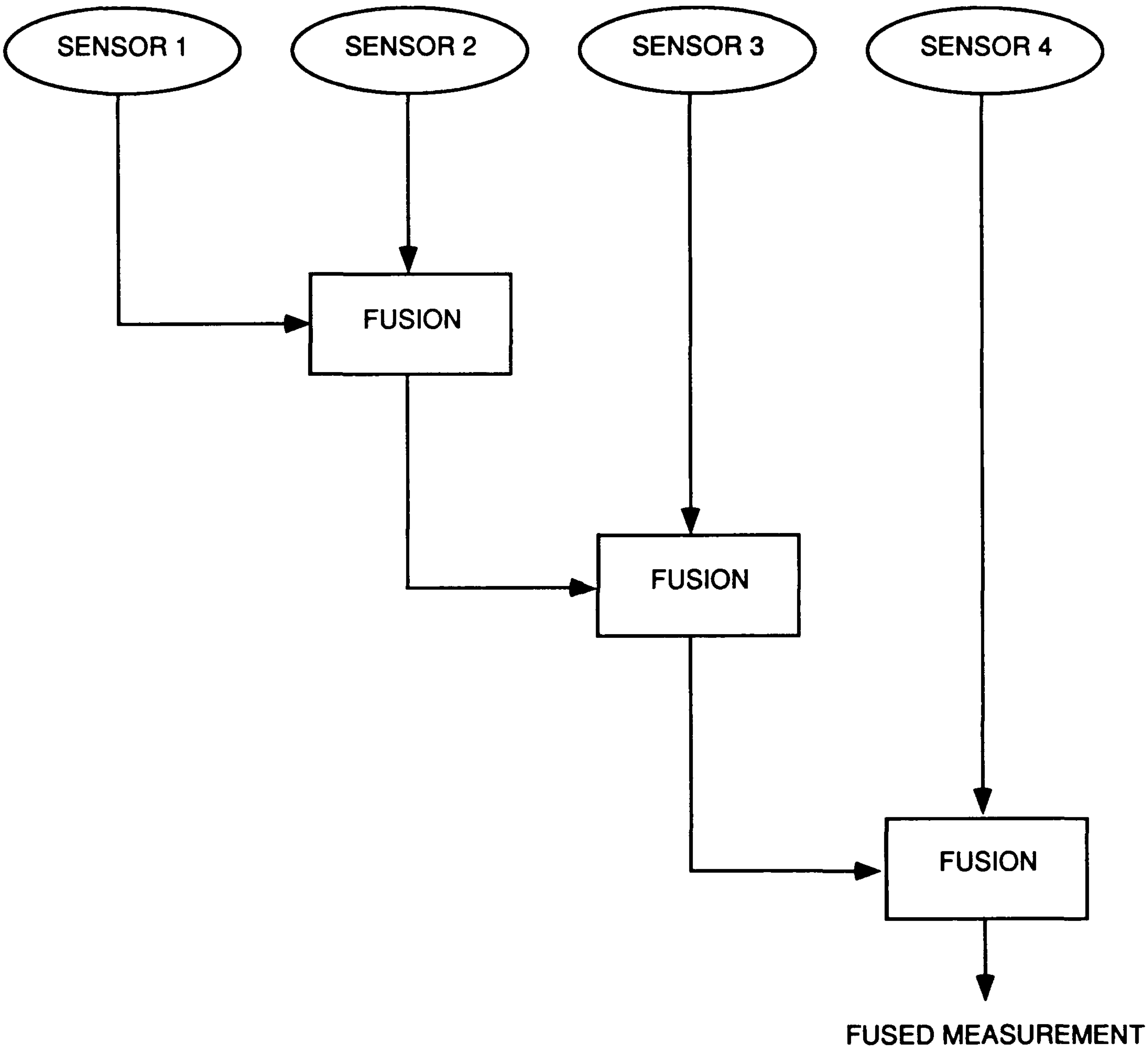


Figure 2.3-4 Fusion Tree for the HDL/CIP Measurement Preprocessing

	AVAILABLE MEASUREMENTS	SENSORS INVOLVED IN FUSION
MEASUREMENT TYPE 1	x-pos, y-pos	THERMAL IMAGER
MEASUREMENT TYPE 2	x-pos, y-pos, z-pos	AIR RADAR ONLY OR THERMAL IMAGR AND AIR RADAR
MEASUREMENT TYPE 3	x-pos, y-pos, x-vel, y-vel	GROUND RADAR ONLY OR GROUND RADAR AND THERMAL IMAGER
MEASUREMENT TYPE 4	x-pos, y-pos, z-pos, x-vel, y-vel	AIR RADAR AND GROUND RADAR OR AIR RADAR AND GROUND RADAR AND THERMAL IMAGER

Table 2.3-4 Measurement Type Determined by the Sensors Involved in Fusion

2.4. Maneuvering Targets and Multiple Model Approach

A great deal of attention has been focused on the problem of tracking maneuvering targets, and several approaches have been proposed. These vary from a simple method like adjusting process noise covariance to a complex method such as the Interacting Multiple Model (IMM) approach. More details can be found in [Fortman, Bar-Shalom].

Since the MMAS simulation data shows an example of a highly maneuvering target, see figure 2.4-1 obtained from the file iew_d3e1, implementation of a suitable maneuvering target tracking algorithm for the HDL/CIP is necessary. Among the well known algorithms, the IMM method may provide rather good performance with efficient computation for the HDL/CIP. During the implementation of the IMM, the following questions are considered.

1) How can we integrate the IMM with the SRIF? And how can we implement data association, especially, measurement-to-track association, in this environment?

2) What kind of target models are suitable for the HDL/CIP ?

Discussions about the second question are deferred until section 2.7.1. In this section, only the first question is considered.

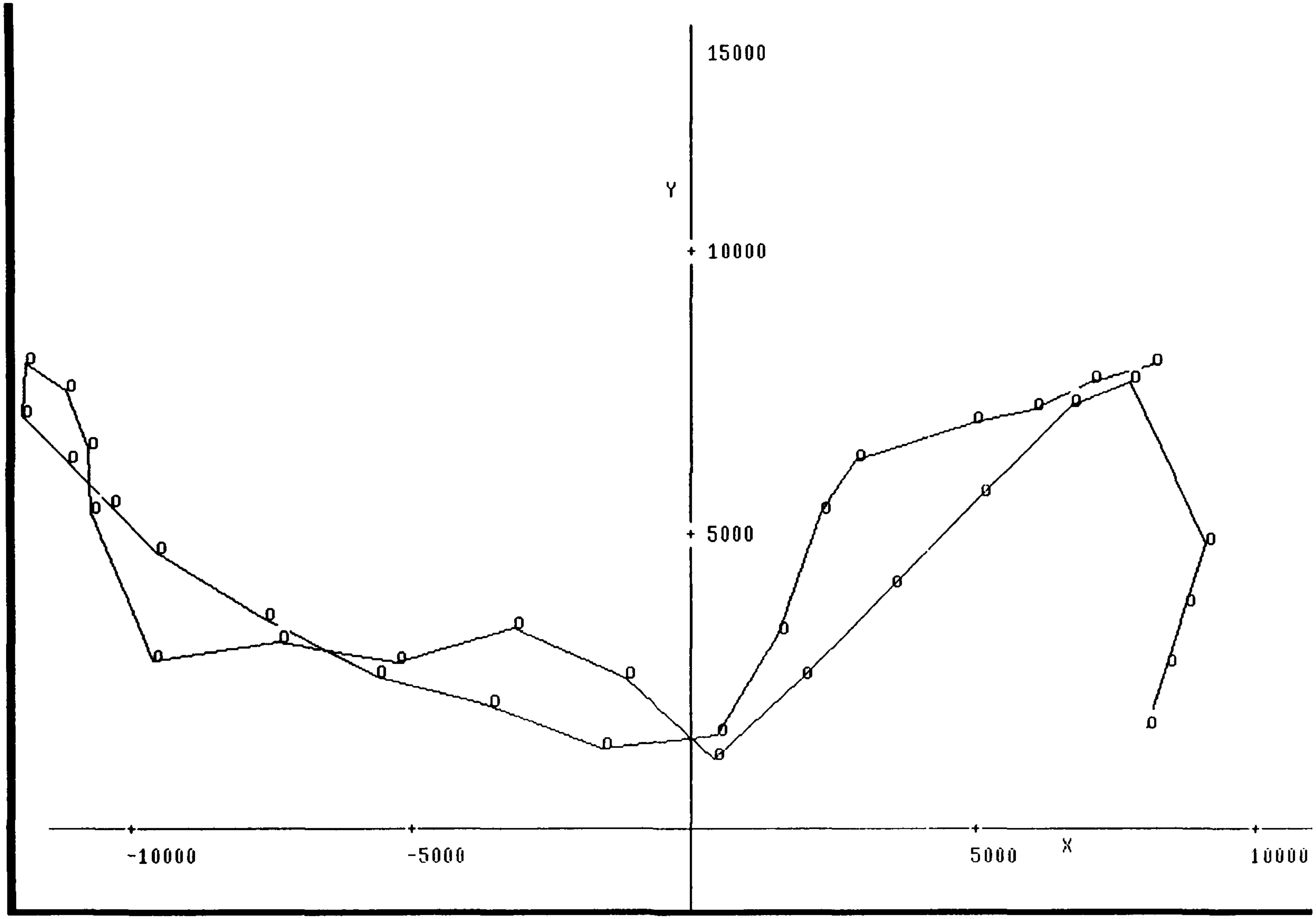


Figure 2.4-1 Example of Highly Maneuvering Target Trajectory

INTEGRATION OF THE IMM WITH THE SRIF: The following steps describe the one cycle of the IMM algorithm for a system which consists of r SRIFs.

step 0:

Set initial parameters. The initial parameters include model switching transition matrix (p_{ij}), $i,j = 1, \ldots, r$, initial model probabilities $\mu_j(0)$, initial state and corresponding covariance, $\hat{x}^j(0|0)$ and $Q^j(0|0)$ for each filter $j = 1,\ldots,r$.

step 1:

Mixing of state estimates and covariances. First, compute

$$\bar{c}_j = \sum_{i=1}^{r} p_{ij}\mu_i(k-1) \quad \text{and} \tag{2.4-1}$$

$$\mu_{i|j}(k-1|k-1) = \frac{1}{\bar{c}_j} \cdot p_{ij}\mu_i(k-1) \tag{2.4-2}$$

Then mixed state and covariance of the filter j is obtained by

$$\hat{x}^{0j}(k-1|k-1) = \sum_{i=1}^{r} \hat{x}^i(k-1|k-1)\mu_{i|j}(k-1|k-1) \quad \text{, and} \tag{2.4-3}$$

$$Q^{0j}(k-1|k-1) = \sum_{i=1}^{r}(Q^i(k-1|k-1) + (\hat{x}^i(k-1|k-1) - \hat{x}^{0j}(k-1|k-1))$$
$$(\hat{x}^i(k-1|k-1) - \hat{x}^{0j}(k-1|k-1))^t)\mu_{i|j}(k-1|k-1) \tag{2.4-4}$$

Sometimes, dimensions of state variables of filters are different, for example, dimensions of state variables of 3-dimensional constant velocity and constant acceleration models are 6 and 9, respectively. Then, to have the same dimensional vectors and matrices in (2.4-3) and (2.4-4), it is necessary to extend 6 dimensional vector to 9 dimensional vector and 6×6 matrix to 9×9 matrix. This can be achieved by augmenting the components of vector and entries of matrices corresponding to x-acceleration, y-acceleration, and z-

acceleration components, which are set to zeros.

step 2:

Get input information matrix $R_{k-1}^j(+)$ and information vector $z_{k-1}^j(+)$ for the filter j. First, the original size of state estimate vector and covariance matrix should be obtained from the mixed state estimate and the mixed covariance. Then, by applying the Cholesky decomposition algorithm, get the information matrix as described in section 2.2.2.1. The information vector also can be obtained in the same manner as described in that section.

step 3:

Perform the time update step for each filter to get $R_k^j(-)$ and $z_k^j(-)$.

step 4:

Perform the measurement update step for each filter to get $R_k^j(+)$ and $z_k^j(+)$. From $R_k^j(+)$ and $z_k^j(+)$, recover the filtered state estimate $\hat{x}^j(k \mid k)$ and its covariance $Q^j(k \mid k)$ for each filter j. The recovery process is the inverse of the factorization process used to get the information matrix and the information vector.

step 5:

Get the likelihood values Λ_j from each filter j. Since the measurement update has been completed for each filter, an evaluation method utilizing the SRIF variables is more suitable here.

step 6:

Update model probabilities. First, compute

$$c = \sum_{i=1}^{r} \Lambda_i(k)\bar{c}_i \ .$$

Here $\bar{C}_i$ is given by (2.4-1). Then the probability for the model or filter j is given by

$$\mu_j(k) = \frac{1}{c} \cdot \Lambda_j(k)\bar{c}_j \ .$$

step 7:

Get the combination of the model-conditioned estimates and covariances as follows. As in step 1, some filtered state estimates should be augmented. Then, the following equations give the fused filtered state estimate and its corresponding covariance.

$$\hat{x}(k \mid k) = \sum_{i=1}^{r} \hat{x}^i(k \mid k)\mu_i(k) \ ,$$

$$Q(k \mid k) = \sum_{i=1}^{r} (Q^i(k \mid k) + (\hat{x}^i(k \mid k) - \hat{x}^{0j}(k \mid k)) \ ,$$
$$(\hat{x}^i(k \mid k) - \hat{x}^{0j}(k \mid k))^t)\mu_i(k) \ .$$

step 8:

Using the augmented filtered state and covariance obtained in step 7, go back to step 1 to resume filtering process.

To test the above algorithm, a target moving in the $X-Y$ plane has been considered. The initial position of the target is (0, 0) and it moves along the Y-axis with constant speed 10 m/sec. For the time period [40, 60], it accelerates with a_x = 0.075 m/sec^2 and a_y = - 0.075 m/sec^2. The it resumes constant motion. It accelerates at the time period [120, 140] again with a_x = 0.075 m/sec^2 and a_y = - 0.075 m/sec^2. Figure 2.4-2 shows this simulated target trajectory.

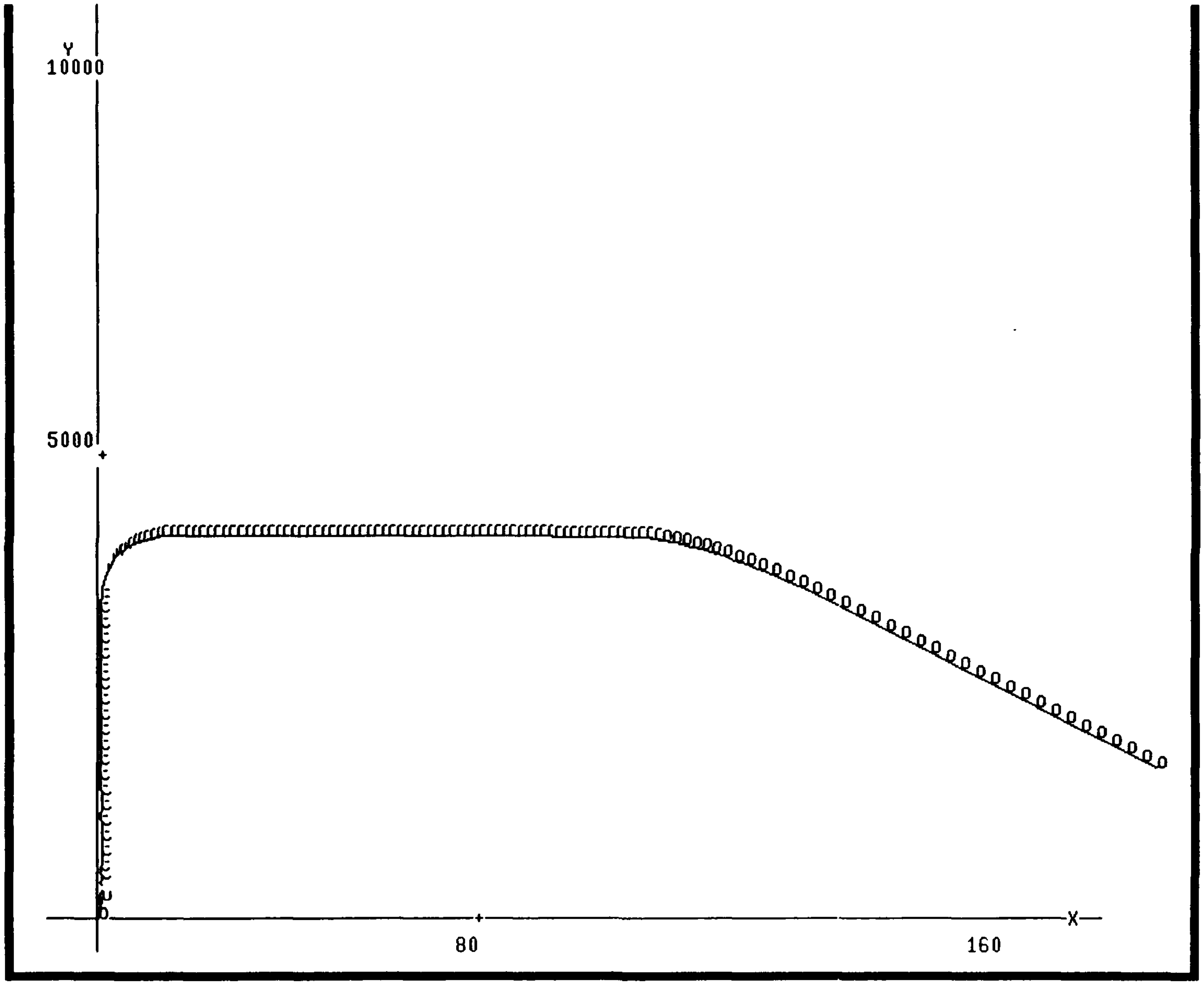


Figure 2.4-2 Simulated Target Trajectory

Two models of the target were used, a 2-dimensional constant velocity model and a 2-dimensional constant acceleration model. The associated Markovian model transition probabilities and model initial probabilities are given in tables 2.4-1 and 2.4-2.

	cv 2	ca 2
cv 2	0.95	0.05
ca 2	0.05	0.95

Table 2.4-1 Model Transition Probabilities

cv 2	ca 2
0.5	0.5

Table 2.4-2 Model Initial Probabilities

In figures 2.4-3 and 2.4-4, the changes of model probabilities of constant acceleration model (ca 2) and constant velocity model (cv 2) are given respectively. As can be seen in figure 2.4-3, when a target is in acceleration mode, the model probability is a very high value (close to 1), and when a target resumes its constant motion, the model probability is a very low value (close to 0). The reverse explanation can be applied to figure 2.4-4. Figure 2.4-5 shows the rms position error of filtered state trajectory from the trajectory in figure 2.4-2.

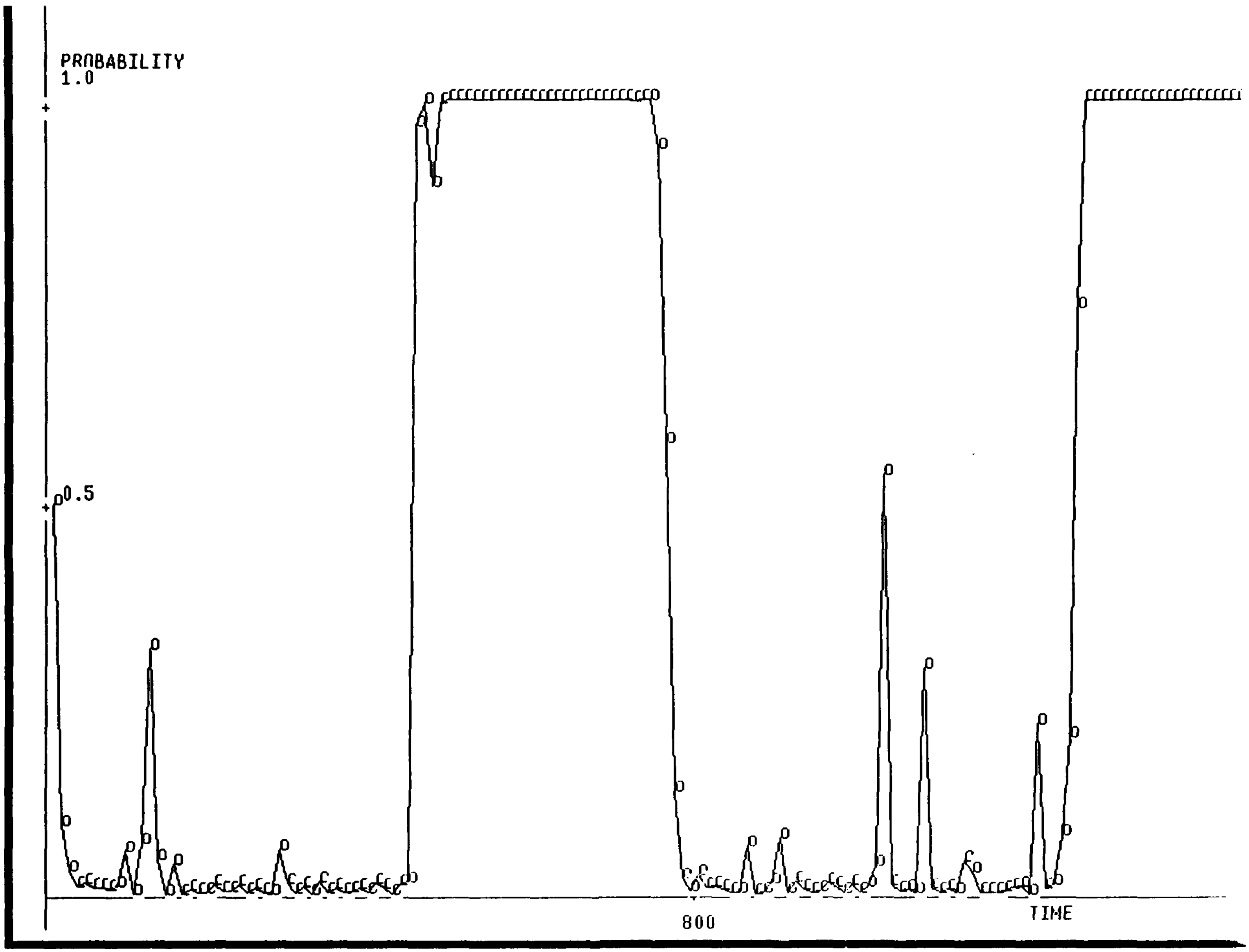


Figure 2.4-3 Change of Model Probability of ca2

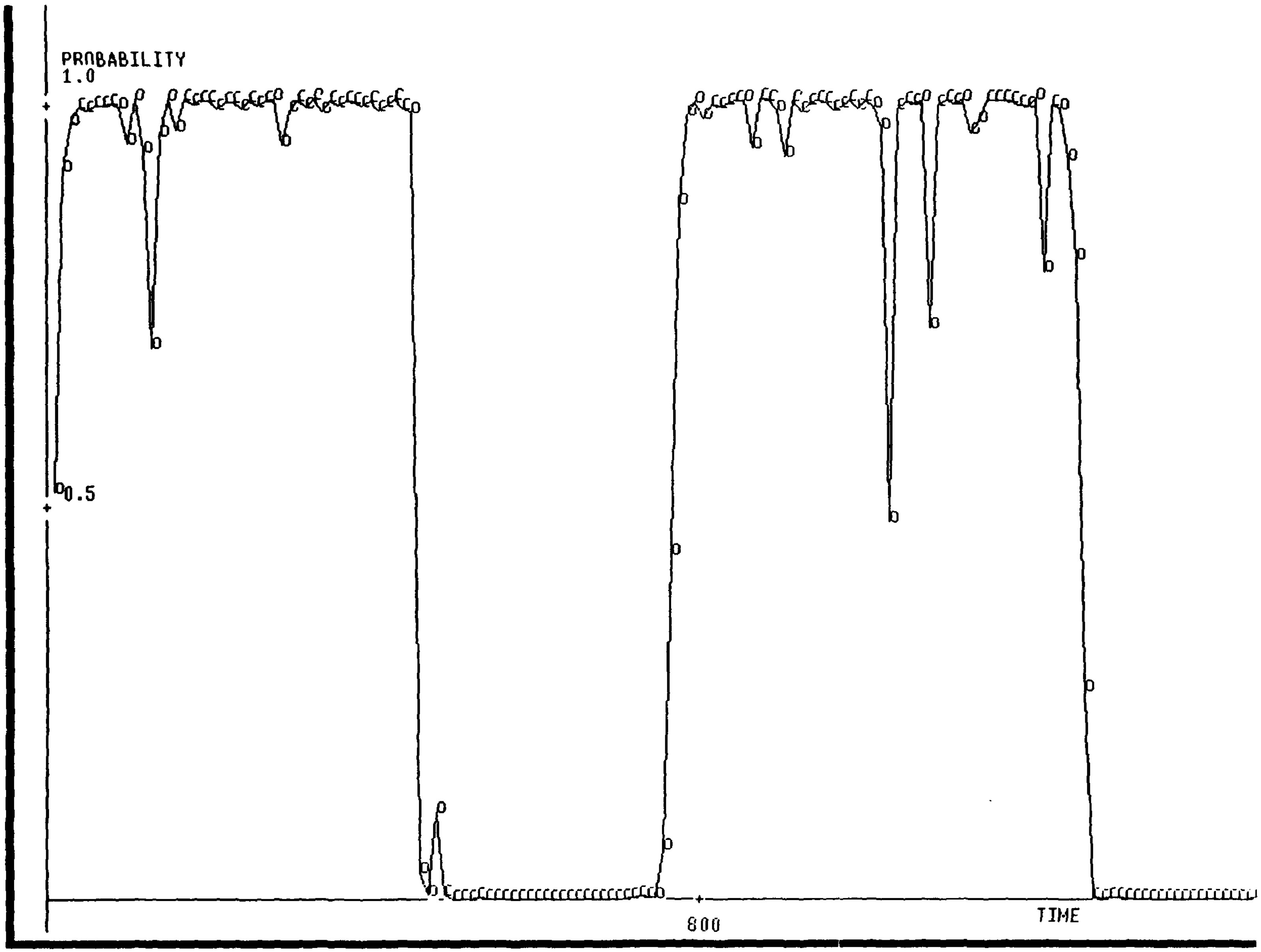


Figure 2.4-4 Change of Model Probability of cv2

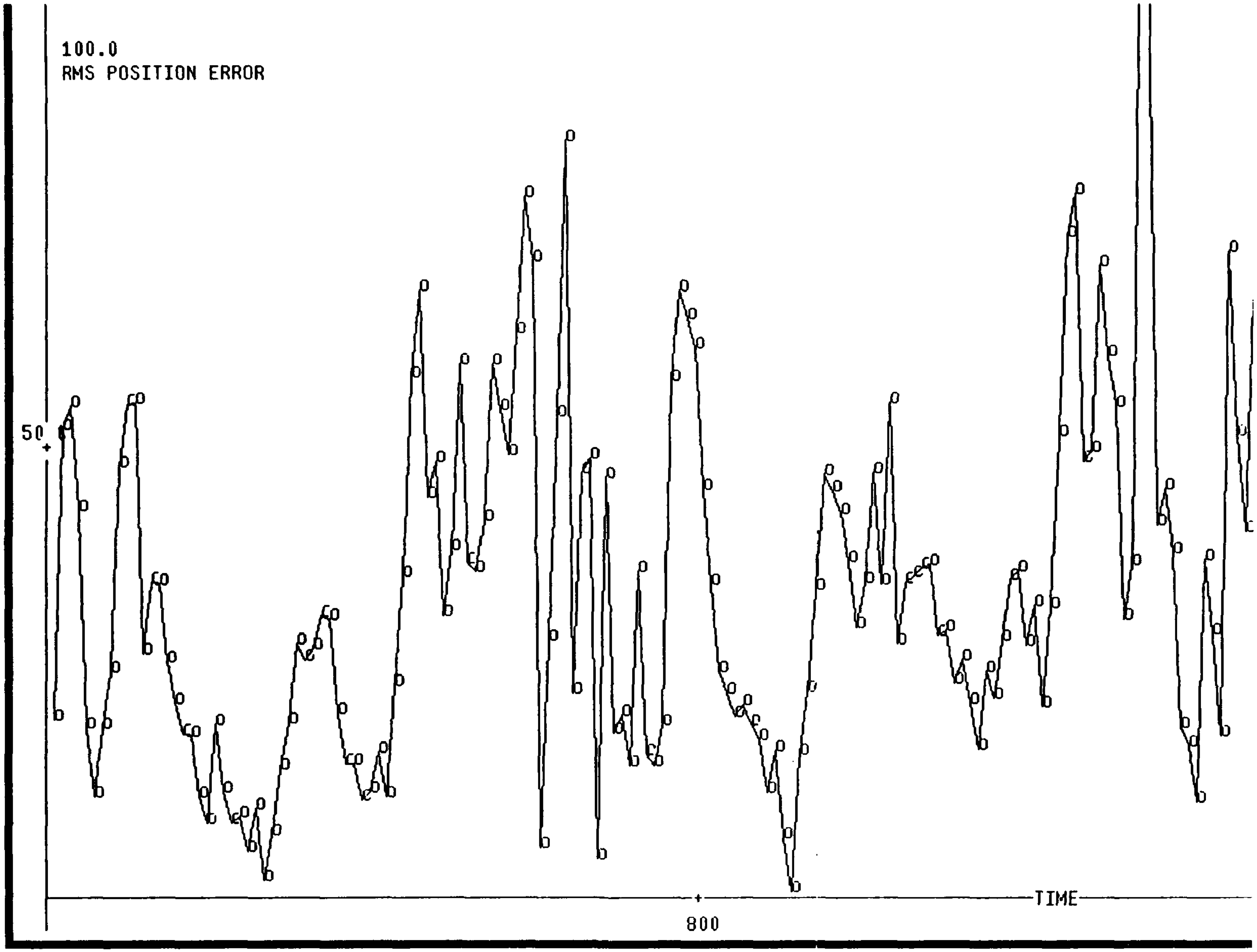


Figure 2.4-5 RMS Value beyween Simulated and Filtered State Trajectories

REVIST TO DATA ASSOCIATION: As described in step 2 in section 2.2.2.2, to associate a target with a measurement, a predicted state estimate is required. Since multiple filters are utilized and each of filter provides its own predicted state estimate, it it necessary to fuse predicted state estimates to get one representative predicted state estimate. That is, by utilizing the following equations

$$\hat{x}(k \mid k-1) = \sum_{i=1}^{r} \hat{x}^i(k \mid k-1)\mu_i(k-1) \tag{2.4-5}$$

$$Q(k \mid k-1) = \sum_{i=1}^{r} (Q^i(k \mid k-1) + (\hat{x}^i(k \mid k-1) - \hat{x}^{0j}(k \mid k-1))$$
$$(\hat{x}^i(k \mid k-1) - \hat{x}^{0j}(k \mid k-1))^t)\mu_i(k-1) \tag{2.4-6}$$

fused predicted state estimate and its covariance are obtained. Note that the model probabilities are the one step previous one, which is not updated yet.

Once the fused predicted state estimate for one target is obtained, the data association procedure resumes as described in section 2.2.2.2.

In summary, we have

step 0:

By utilizing equations (2.4-5) and (2.4-6), get the fused predicted state estimate, and then apply the data association steps described in section 2.2.2.2.

2.5. Low Sampling Rates and Track Update Without Measurements

Usually it is believed that the quality of filtering is dependent on the sampling rate of measurement. If the measurement data are collected periodically with short time interval, track can be formed even without filtering. Sometimes, however, the sensors cannot report the detection of certain targets because the targets are outside of their scanning volumes or for other reasons. Even in these cases, sometimes it is necessary to maintain the tracks of certain targets. See table 2.5-1, which is obtained from the MMAS data file iew_d2e1.

In this section, an approach which updates tracks even without measurements is considered in the SRIF environment. The basic principle behind this is already well known in the estimation area. By performing a time update, a predicted state estimate is obtained. If a new measurement is not available, it is plausible to use the predicted state estimate as a filtered state estimate, and to resume filtering process by starting a time update.

The idea described above can be easily implemented in the SRIF. In the measurement update, substituting a zero matrix and a zero vector where $R_v H$ and $R_v y$ were, then we get

$$T \begin{bmatrix} R_{k+1}(-) & z_{k+1}(-) \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} R_{k+1}(+) & z_{k+1}(+) \\ 0 & e_{k+1} \end{bmatrix} \tag{2.5-1}$$

and $R_{k+1}(+) = R_{k+1}(-)$ and $z_{k+1}(+) = z_{k+1}(-)$, since $R_{k+1}(+)$ is an upper triangular matrix and T is a Householder transformation.

To achieve continuous filtering, the nonblocking network receiving function, netGETANY_NBLK, has been developed. This function checks the message buffer and resume track updating process whether there are any messages in the buffer or not. If there is any message, then the process measurement_update_with_measurement is invoked. Otherwise, the process measurement_update_without_measurement is utilized.

sensor id	time reported	time detected
5003	40.1	40.1
5003	40.3	40.3
2001	43.0	43.0
1001	43.0	43.0
4001	45.0	45.0
4002	45.0	45.0
1001	46.0	46.0
4001	47.5	47.5
4002	47.5	47.5

Table 2.5-1 Example of Low Sampling Rate

2.6. Track Management

In this section, track initiation and track deletion are considered as a part of the track management process. Whenever a measurement is regarded as a new one, a corresponding track file is created and tracking process starts. In section 2.6.1, a detailed description about creating a new track file is provided. When a target remains out of sight of the sensors for a sufficient time, the track file for this target is eliminated. A simple but reasonable method is described in section 2.6.2.

2.6.1. Track Initiation

Track initiation is the process which creates a track file and stores relevant data, such as kinematic data, into the track file. This process is invoked at two places in tracking process. When the first set of measurement data arrives at the CIP, each of them is regarded as a potential track initiator. The other case is when data association is finished. While scanning the measurement data set, if measurements which are not associated with any currently existing tracks are found, these are regarded as measurements of new targets, and track initiation is invoked.

Initiation starts by creating a track file for each of the new measurements, and then determining the initial state and the corresponding initial state error covariance.

The simplest method to determine these kinematic data is based on the finite difference method. For each target, three consecutive position measurements should be accumulated. Time periods between the first and the second, and between the second and the third are necessary. To get a velocity estimate, the second position measurement is subtracted from the third, and the difference is divided by the time period between the second and the third. In a similar way, one step previous velocity estimate can be obtained by utilizing the first and the

second position measurements. To get an acceleration estimate, one step previous velocity estimate is subtracted from the velocity estimate obtained, and the difference is divided by the time period between the first and the second. In this simple manner, we might be able to estimate velocity and acceleration of each target.

However, it is easy to see that the above approach cannot be applied to the HDL/CIP case directly. The reason is that arrival time of measurements is random so that we cannot predict the time interval between one measurement and the next measurement. Instead of using the finite difference method, the following approach is implemented.

Table 2.3-4 shows a general measurement data format which is sent by the measurement preprocessing module to the tracker module. Note also that, table 2.3-4 shows different possible types which a general measurement data format can take, according to what kind of sensors are utilized in measurement fusion step.

Once a measurement is received by the tracker module, it is stored in a buffer temporally, and then copied into the measurement data structure for the tracking process. At that time, each measurement is labeled as a new measurement. See table 2.6.1-1 for the measurement data structure. More details can be found in Volume 2, Part A, section 2.3.2.

Kinematic information stored in the measurement data structure is to be used to initialize track. However, as mentioned in section 2.4, there are 4 different models implemented for the CIP tracker. These are 2-dimensional constant velocity and acceleration models, and 3-dimensional constant velocity and acceleration models. Initialization for each model consists of the following three steps.

step 1:

According to the model type and the measurement type, copy appropriate kinematic data

into the track file from the measurement data structure. Information not available from the measurement data structure, i.e., z-velocity, x-acceleration, y-acceleration, and z-acceleration are set to zero if they are required in the model. Table 2.6.1-2 shows details according to the different cases.

step 2:

According to the model type and measurement type, determine an appropriate initial state error covariance matrix, which takes the form of a diagonal matrix under the assumption that components are independent each other. If any component is actually available from the measurement data structure (which can be determined by measurement type), the corresponding variance is set to a reasonably small number. Otherwise, i.e., a component of the state variable which is required but not available in the measurement, then the corresponding variance is set to a reasonably big number. Table 2.6.1-3 shows details.

step 3:

Initialize each model by following the procedures described in section 2.2.2.1 to start filtering.

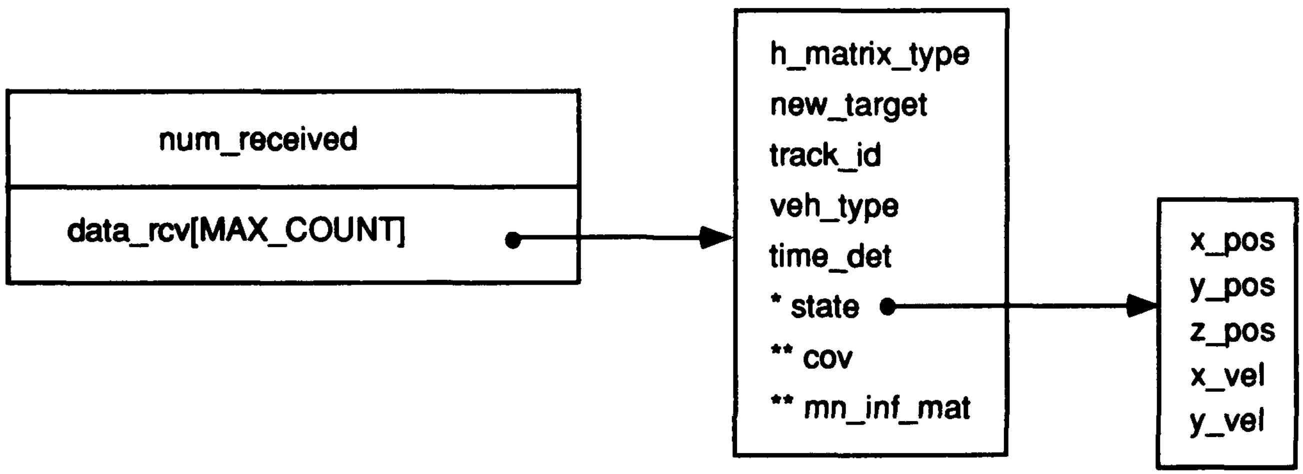


Table 2.6.1-1 Measurement Data Structure

	MEAS TYPE 1	MEAS TYPE 2	MEAS TYPE 3	MEAS TYPE 4
CV2	copy x, y set x-vel, y-vel = 0	copy x, y set x-vel, y-vel = 0	copy x, y, x-vel, y-vel	copy x, y, x-vel, y-vel
CA2	copy x, y set x-vel, y-vel, x-acc, y-acc = 0	copy x, y set x-vel, y-vel, x-acc, y-acc = 0	copy x, y, x-vel, y-vel set x-acc,y-acc = 0	copy x, y, x-vel, y-vel set x-acc,y-acc = 0
CV3	copy x, y set z, x-vel, y-vel, z-vel = 0	copy x, y, z set x-vel, y-vel, z-vel = 0	copy x, y, x-vel, y-vel set z, z-vel = 0	copy x, y, z, x-vel, y-vel set z -vel = 0
CA3	copy x, y set z, x-vel, y-vel, z-vel, x-acc, y-acc, z-acc = 0	copy x, y, z set x-vel, y-vel, z-vel, x-acc, y-acc, z-acc = 0	copy x, y, x-vel, y-vel set z, z-vel, x-acc, y-acc, z-acc = 0	copy x, y, z, x-vel, y-vel set z-vel, x-acc, y-acc, z-acc = 0

Table 2.6.1-2 Initial State Assignment

	MEAS TYPE 1	MEAS TYPE 2	MEAS TYPE 3	MEAS TYPE 4
CV2	x-vel, y-vel	x-vel, y-vel		
CA2	x-vel, y-vel, x-acc, y-acc	x-vel, y-vel, x-acc, y-acc	x-acc,y-acc	x-acc,y-acc
CV3	z, x-vel, y-vel, z-vel	x-vel, y-vel, z-vel	z, z-vel	z-vel
CA3	z, x-vel, y-vel, z-vel, x-acc, y-acc, z-acc	x-vel, y-vel, z-vel, x-acc, y-acc, z-acc	z, z-vel, x-acc, y-acc, z-acc	z-vel, x-acc, y-acc, z-acc

Table 2.6.1-3 Entries of Covariance Matrix with Big Number

2.6.2. Track Deletion

Track deletion is achieved by eliminating corresponding track file from the list of currently existing track files. The problem is to decide when a track file should be deleted. The first case is when a target is outside of the scanning volumes of the sensors for a while. The second case is when the operator of the CIP decides to delete certain tracks.

The current CIP tracker software accepts only the first case. To achieve this, a simple, efficient method is implemented. Whenever a set of measurements are received, tracks are updated based on those measurements. After the data association procedure, it may happen that some tracks are not associated with any received measurements. That means that the corresponding targets might be out of sight of the sensors, and they become candidates for deletion.

However, measurements may be absent because of faulty sensors. To minimize the risk of deletion of targets which are still inside of the scanning volumes of some sensors, the number of track updates without measurement is accumulated whenever it happens. If the accumulation number exceeds the threshold, then the corresponding track is deleted. A track survives if it is associated with a measurement again and the accumulated number is less than the threshold. In this case, the accumulated number should be reset to zero.

As discussed in section 2.5, track updating is based on the nonblocking network receiving function to continue updating, even though there are no measurements. In this case, tracks of all airborne targets are updated without measurements and the numbers are not counted.

2.7. Other Issues

In this section, topics such as the selection of kinematic models relevant to the vehicles to be tracked, the robustness of the SRIF with respect to initial parameters, a vehicle type adjustment based on filtered state estimate, and the menuing system are described.

2.7.1. Different Types of Targets and Modeling

As described in section 2.1, the MMAS simulation data contains information about airborne targets and ground targets. This means that the HDL/CIP should be equipped with the capability of tracking both types of targets. In addition to the measurement sampling rates, another important factor which determines the filtering quality is the models selected for each target. In this section, models selected for the HDL/CIP are detailed.

According to the information in the MMAS data, ground targets moved at constant speed except in rough terrain, where they slowed down. Airborne targets also flew at a constant speed. This information suggests to use 2-dimensional constant velocity kinematic model for ground targets and 3-dimensional constant velocity model for airborne targets.

However, as seen in section 2.4, the maneuvering characteristics of targets become an important issue in tracking tasks. To adjust the direction of movement, each target should have the capability of acceleration and deceleration. This also leads to the use of 2-dimensional and 3-dimensional constant acceleration kinematic models.

Each of the following models is a typical form which represents one of the above cases, and takes the form of

$$x(k+1) = Fx(k) + Gw(k) .$$

For notational convenience, let's define the following matrices.

$$A = \begin{bmatrix} 1 & \Delta T & \Delta T^2/2 \\ 0 & 1 & \Delta T \\ 0 & 0 & 1 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix}$$

$$C = \begin{bmatrix} \Delta T^2/2 \\ \Delta T \\ 1 \end{bmatrix}$$

$$D = \begin{bmatrix} \Delta T^2/2 \\ \Delta T \end{bmatrix}$$

Here ΔT is the time difference between one step and the next step of filtering.

1) 3-dimensional constant velocity kinematic model: The state variables consist of x, $\dot{x}$, y, $\dot{y}$, z, and $\dot{z}$, and the noise terms are w_x, w_y, and w_z. The F and G matrices are given by

$$F = \begin{bmatrix} B & 0 & 0 \\ 0 & B & 0 \\ 0 & 0 & B \end{bmatrix} \quad \text{and} \quad G = \begin{bmatrix} D & 0 & 0 \\ 0 & D & 0 \\ 0 & 0 & D \end{bmatrix}$$

2) 3-dimensional constant acceleration kinematic model: The state variables consist of x, $\dot{x}$, $\ddot{x}$, y, $\dot{y}$, $\ddot{y}$, z, $\dot{z}$, and $\ddot{z}$. The noise terms are w_x, w_y, and w_z, F and G are given as follows.

$$F = \begin{bmatrix} A & 0 & 0 \\ 0 & A & 0 \\ 0 & 0 & A \end{bmatrix} \quad \text{and} \quad G = \begin{bmatrix} C & 0 & 0 \\ 0 & C & 0 \\ 0 & 0 & C \end{bmatrix}$$

3) 2-dimensional constant velocity kinematic model: The state variables consist of x, $\dot{x}$, y, and $\dot{y}$. The noise terms are w_x and w_y, and F and G are as follows.

$$F = \begin{bmatrix} B & 0 \\ 0 & B \end{bmatrix} \quad \text{and} \quad G = \begin{bmatrix} D & 0 \\ 0 & D \end{bmatrix}$$

4) 2-dimensional constant acceleration kinematic model: The state variables are x, $\dot{x}$, $\ddot{x}$, y, $\dot{y}$, and $\ddot{y}$. The noise terms are w_x and w_y. F and G are given by

$$F = \begin{bmatrix} A & 0 \\ 0 & A \end{bmatrix} \quad \text{and} \quad G = \begin{bmatrix} D & 0 \\ 0 & D \end{bmatrix}$$

Sometimes, the movement of targets is limited by terrain. This limitation becomes more severe when ground targets are considered. To avoid this limitation, instead of developing more complicated models for ground targets, developing a system which can combine the terrain information with filtering results to generate more reliable track information might be a plausible approach.

2.7.2. Multiple Filter Initialization

Even though it has been proved that the SRIF is numerically more stable than the conventional Kalman filter, the robustness of the SRIF with respect to some parameters still raises some interesting questions whenever the implementation is considered.

To implement the SRIF, the parameters required are, as mentioned in section 2.2.2.1, the initial state and its error covariance, the process noise mean vector and process noise error covariance, and the measurement noise error covariance. Measurement noise mean vector is usually assumed to be zero.

As described in section 2.6.1 (see table 2.6.1-3), some entries of initial state error covariance matrix are assigned a reasonably big number when the corresponding components of the initial state are not available. Since information matrix is defined as an inverse of the square root of the covariance, the entries of covariance with big numbers give an information matrix with entries of small numbers. These small numbers sometimes lead to numerical underflow or overflow. Similar effects are also expected from the process noise covariance.

The above arguments imply that a robust implementation of the SRIF requires the estimation of the levels of initial, process and possibly measurement noise for an individual target. That is, it is necessary to determine boundaries for the numbers to be used to initialize filter.

For the HDL/CIP tracking system, since most of the design is based on the MMAS simulation data, these numbers are determined by testing only. However, to be implemented in real situation, a more systematic and sophisticated method should be incorporated into the procedure to determine these parameters. For example, a maximum likelihood approach to parameter estimation is a very general method which has been applied to the problem of

determining the parameters of a linear dynamical models described in the previous section.

2.7.3. Adjustment of Vehicle Type

Even without measurement data, the HDL/CIP tracking system is designed to proceed to update tracks. If all tracks, whether they are for ground targets or airborne targets, are updated without measurements, it may impose a heavy computational load on the system. Since the speeds of ground targets are slow compared to those of airborne targets, it is plausible to update only airborne targets. Hence, it is necessary to distinguish vehicle types.

The method currently implemented to distinguish vehicle types is very simple. It utilizes the x-velocity and y-velocity of filtered state estimate of each target. Every time a track is updated with a measurement, speed of the vehicle is computed and compared with two threshold values, such as 40 m/sec and 20 m/sec, which can be decided by the operators. If the speed of the vehicle exceeds the high speed threshold, then vehicle type is adjusted as an airborne target. If the speed is below the lower threshold, the vehicle is regarded as a ground target. Otherwise, it is a vehicle with unknown type.

The procedure described above contains some risks because it is based on the kinematic information only. To make a more reliable decision about vehicle type, more information, which is not necessarily kinematic or numerical, from the sensors is required. Also, an algorithm is needed to fuse the information, which may contain both the numerical and non-numerical types, to get more reliable decisions.

2.7.4. Interrupt Handling

The handling of interruptions plays an important part in the design of the HDL/CIP tracker software.

In the middle of the tracking process, an interruption might be necessary to improve tracking performance by changing the tracker parameters listed in section 2.3.3, Volume 2, Part A.

The steps to interrupt the tracker and to resume tracking process are described in section 9, Volume 3.

Once an interruption signal is received by the tracker, the previously processed information will be lost, but the communication network continues to function. Hence, the messages are continuously received. The tracker process restarts with the track initiation process based on the first measurements queued in message buffer.

This design approach is plausible since tracker itself does not need to keep track histories for all targets. As tracks are updated, the updated results are sent to database and then are lost as the next update is processed.

3. Performance Evaluation Test

This section includes test results of the HDL/CIP tracker software on the MMAS simulation data. Section 3.1 contains descriptions how the tests are performed. In section 3.2, test results from the important individual modules are described. In the last section, 3.3, test results for some of the IEW data files are provided.

3.1. Description of Testing Environment

A SunSPARC 330 has been used as the main development and test machine. As described in Volume 2, Part A, the design of the HDL/CIP tracker software is based on the CIP network software. It is required that the network software run as a background on the system. Once the network software is running on the machine, the four tracking tasks should be run by utilizing the multi-task capability of the SUN operating system. These four tasks consist of send_meas, ppm_dri[illegible]n, and tracker2. Detailed explanations about each of these tasks are included in Volume 2, Part A. Also, Volume 3 provides about how to set up the system.

3.2. Test Results

In this section, the test results of the HDL/CIP tracker are described. The test file is iew_d2e2, which contains simulated trajectories of two A-7's. The measurement noise covariances for airborne radar, ground radar, and thermal imager are given below.

100.0	0.0	0.0	0.0	0.0
	100.0	0.0	0.0	0.0
		100.0	0.0	0.0
			1000000.0	0.0
				1000000.0

100.0	0.0	0.0	0.0	0.0
	100.0	0.0	0.0	0.0
		1000000.0	0.0	0.0
			100.0	0.0
				100.0

100.0	0.0	0.0	0.0	0.0
	100.0	0.0	0.0	0.0
		1000000.0	0.0	0.0
			1000000.0	0.0
				1000000.0

Two models cv3 and ca3 are selected. The initial model probabilities are given by 0.5 and 0.5 and the model transition matrix is given by

0.95	0.05
0.05	0.95

The following are the process noise mean vectors and the process noise covariances for cv3 and ca3, respectively.

0.0	0.0	0.0
1000.0	0.0	0.0
	1000.0	0.0
		1000.0

0.0	0.0	0.0
1000.0	0.0	0.0
	1000.0	0.0
		1000.0

Other parameters utilized are defined as follows. A description for each parameter is included in section 2.3.3, Volume 2, Part A.

MaxNumModel: 5

MAX_COUNT: 50

TRACK_DEL_THRESHOLD: -5

MAX_AIRVEH_SPEED: 300

MAX_GRVEH_SPEED: 20

POS_COV: 1.0e+5

VEL_COV: 1.0e+4

ACC_COV: 1.0e+3

SM_COV: 1.0e+2

DELAY: 2.0

CHI_VAL: 20000.0

MAX_MEAS: 50

DATA_ACCUMULATION_INTERVAL: 0.5

MAX_MEAS: 50

DIM: 5

.

First, table 3.2-1 shows part of the output from the read_iew program, which is used as an input to the ppm module. Table 3.2-2 contains three message buffers from the module ppm. Table 3.2-3 and 3.2-4 are tracking outputs from tracker module.

The first measurement from the sensor 5003 (ground radar) arrived at time 20.1 with a detection time 20.1. Since there is only one measurement, no fusion of measurements happens in ppm module. x-pos, y-pos, x-vel, and y-vel are copied into the message buffer and z-pos is set to zero since it is not available from ground radar. The corresponding covariance is also obtained. Only the entry which corresponds to z-pos component has a big number compared to other entries. veh_type is set to -1 since it is not available.

In table 3, the first three rows show that initial track has been formed based on the message received from the ppm module. The received measurements are directly used as an initial state.

At the time 23.0, a measurement arrived from sensor 2001, and two measurements were received from sensors 4002 and 4001 at the time 25.0, 2 seconds later. Note that these are from the same target 21. However, the detection time of the measurements from sensors 4001 and 4002 were 0.5 sec earlier than the detection time of 2001. That is, these measurements are out of sequence.

The second message buffer in table 3.2-2 contains just 1 measurement with detection time 22.30. As the header pointed out the fusion process has been performed.

Two interesting things are observed here. First, measurements from 4001 and 4002 were processed before the measurement from 2001 in the ppm module. Second, after clustering, it was decided that these two measurements were from the same source and fusion process was invoked.

The fused x-pos, y-pos, x-vel, and y-vel were stored into message buffer and sent to tracker module. Note the changes in covariance matrix, which shows that entries corresponding to available measurements became less than the original values.

In table 3.2-3, observe the tracking result after the second measurement was received. The number of measurements is 1, which is exactly same as the second message buffer in table 3.2-2.

The next 6 rows contain the predicted positions from models cv3 and ca3. The next 9 rows contain the averaged predicted state of the predicted states from the cv3 and ca3 models.

The next line shows that currently there is only one target in track file list and only one measurement has been received. The value 9.870757e+01 is corresponding log-likelihood value.

Since there is only one track and one measurement, the Munkres' algorithm gives automatic association, which is represented by assign[0]: O. Here, assign[0] represents the first track and 0 represents the measurement with index 0.

Using the associated measurement, a measurement update is performed. Then the likelihood values can be obtained by the method based on the SRIF. These likelihood values are used to update the model probabilities.

Finally, from the filtered state estimates from both models, the fused state was obtained and only the x, y, and z positions are listed.

Based on the velocity estimate, the vehicle adjustment process was invoked. Since speed of the vehicle was above the airborne target threshold, the vehicle type was adjusted to 102, which represents an airborne target. The fused positions are printed again in the last line. Note that the z position is still 0.

Similar explanations can be given to the track information in table 3.2-4, where the measurement from sensor 2001 was utilized.

sensor unit vehicle time_true time_rep time_det

x y z x_err y_err z_err

v_type v_speed v_dir x_sig y_sig z_sig

==

5003 1 21 20 20.1 20.1

6843.3 -9826.9 1117.2 0.7 -99.1 547.3

102 240.9 265. 1.0 53.3 1486.5

2001 1 21 20 23.0 23.0

6864.0 -9747.6 1631.1 -20.0 -178.4 33.4

102 240.9 265. 16.1 128.7 57.8

4002 1 21 20 25.0 22.5

6855.1 -10024.3 1758.9 -11.1 98.3 -94.4

102 240.9 265. 29.5 145.2 227.2

4001 1 21 20 25.0 22.5

6829.4 -9776.3 1796.5 14.6 -149.7 -132.0

102 240.9 265. 19.0 149.7 234.0

Table 3.2-1 Input to ppm Module

number of meas: 1
time_det: 20.20
time_limit: 20.41
Htype: 3 veh_type: -1 x_pos: 6843.30 y_pos: -9826.90 z_pos: 0.00
x_vel: -239.98 y_vel: -21.00
COVARIANCE
100.000000 0.000000 0.000000 0.000000 0.000000
0.000000 100.000000 0.000000 0.000000 0.000000
0.000000 0.000000 1000000.000000 0.000000 0.000000
0.000000 0.000000 0.000000 100.000000 0.000000
0.000000 0.000000 0.000000 0.000000 100.000000

BEFORE FUSE next->t_detected: 22.500000

number of meas: 1
time_det: 22.30
time_limit: 22.56
Htype: 3 veh_type: -1 x_pos: 6842.25 y_pos: -9900.30 z_pos: 0.00
x_vel: -239.98 y_vel: -21.00
COVARIANCE
50.000000 0.000000 0.000000 0.000000 0.000000
0.000000 50.000000 0.000000 0.000000 0.000000
0.000000 0.000000 500000.000000 0.000000 0.000000
0.000000 0.000000 0.000000 50.000000 0.000000
0.000000 0.000000 0.000000 0.000000 50.000000

number of meas: 1
time_det: 23.20
time_limit: 23.42
Htype: 3 veh_type: -1 x_pos: 6864.00 y_pos: -9747.60 z_pos: 0.00
x_vel: -239.98 y_vel: -21.00
COVARIANCE
100.000000 0.000000 0.000000 0.000000 0.000000
0.000000 100.000000 0.000000 0.000000 0.000000
0.000000 0.000000 1000000.000000 0.000000 0.000000
0.000000 0.000000 0.000000 100.000000 0.000000
0.000000 0.000000 0.000000 0.000000 100.000000

Table 3.2-2 Message Buffer from ppm Module

```
meas.num_received: 1
Hmat: 3  x_pos: 6843.299805  y_pos: -9826.900391  z_pos: 0.000000
x_vel: -239.983215  y_vel: -20.996756  time_det: 20.200001  veh_type: -1


meas.num_received: 1
Hmat: 3  x_pos: 6842.250000  y_pos: -9900.299805  z_pos: 0.000000
x_vel: -239.983215  y_vel: -20.996756  time_det: 22.299999  veh_type: -1
CV3.pred_state[0]  6339.334961
CV3.pred_state[2]  -9870.992188
CV3.pred_state[4]  0.000000

CA3.pred_state[0]  6339.335938
CA3.pred_state[3]  -9870.994141
CA3.pred_state[6]  0.000000
IN AVG_PREDICTION: avg_pred_state[0] = 6339.335449
IN AVG_PREDICTION: avg_pred_state[1] = -239.983139
IN AVG_PREDICTION: avg_pred_state[2] = 0.000182
IN AVG_PREDICTION: avg_pred_state[3] = -9870.993164
IN AVG_PREDICTION: avg_pred_state[4] = -20.996841
IN AVG_PREDICTION: avg_pred_state[5] = -0.000266
IN AVG_PREDICTION: avg_pred_state[6] = 0.000000
IN AVG_PREDICTION: avg_pred_state[7] = 0.000000
IN AVG_PREDICTION: avg_pred_state[8] = 0.000000


num_target: 1  num_meas: 1
After LOG_LIKE
9.870757e+01

assign[0]: 0

MODEL_PROBABILITY: model_probability[0] = 0.000008
 CV3.likeihood_val = 1.236640e-08
MODEL_PROBABILITY: model_probability[1] = 0.999992
 CA3.likeihood_val = 1.627030e-03


FUSION STATE(SEND to DISPLAY)
      X = 6762.366211
      Y = -9895.644531
      Z = 0.000000

------------TRACK TRAVERSING------------
track_id: 1
veh_type: 102
associated_meas: 0
FUSION X=  6762.37  Y= -9895.64  Z=     0.00
```

Table 3.2-3 Track Information from Tracker Module

meas.num_received: 1
Hmat: 3 x_pos: 6864.000000 y_pos: -9747.599609 z_pos: 0.000000
x_vel: -239.983215 y_vel: -20.996756 time_det: 23.200001 veh_type: -1
CV3.pred_state[0] 6641.997559
CV3.pred_state[2] -9920.115234
CV3.pred_state[4] 0.000000

CA3.pred_state[0] 6607.040527
CA3.pred_state[3] -9918.078125
CA3.pred_state[6] 0.000000
IN AVG_PREDICTION: avg_pred_state[0] = 6607.041016
IN AVG_PREDICTION: avg_pred_state[1] = -189.969940
IN AVG_PREDICTION: avg_pred_state[2] = -38.640724
IN AVG_PREDICTION: avg_pred_state[3] = -9918.078125
IN AVG_PREDICTION: avg_pred_state[4] = -23.911510
IN AVG_PREDICTION: avg_pred_state[5] = 2.251318
IN AVG_PREDICTION: avg_pred_state[6] = 0.000000
IN AVG_PREDICTION: avg_pred_state[7] = 0.000000
IN AVG_PREDICTION: avg_pred_state[8] = 0.000000

num_target: 1 num_meas: 1
After LOG_LIKE
4.562517e+02

assign[0]: 0

MODEL_PROBABILITY: model_probability[0] = 0.000000
CV3.likeihood_val = 2.797018e-06
MODEL_PROBABILITY: model_probability[1] = 1.000000
CA3.likeihood_val = 3.239890e-01

FUSION STATE(SEND to DISPLAY)
X = 6707.987305
Y = -9842.945312
Z = 0.000000

------------TRACK TRAVERSING------------
track_id: 1
veh_type: 102
associated_meas: 0
FUSION X= 6707.99 Y= -9842.95 Z= 0.00

Table 3.2-4 Track Information from Tracker Module

Table 3.2-5 shows a message buffer with two measurements detected at 80.20. Also, it shows one measurement in message buffer, which is detected at 82.40.

Table 3.2-6 shows that tracker module has received two measurements. Until these two measurements were received, there was only one track in tracker module. The data association result shows that assign[0]: 0. It means that the currently existing track was associated with the measurement with index 0. Then the measurement with index 1 was regarded as a new measurement, and the track initiation process was invoked. In table 3.2-6, only the fused filtered state of the first track was included.

In table 3.2-7, tracking results for two tracks are included. First, note that one measurement was received. As data association results, we have assign[0]: 0, and assign[1]: -1. It means that the first track was associated with the measurement, and the second track which was created in the last step was not associated with the measurement. Then measurement update without measurement was invoked for the second track.

```
number of meas: 2
time_det:   80.20
time_limit:   80.42
Htype: 3  veh_type: -1  x_pos: -2120.50  y_pos: -3391.40  z_pos:     0.00
x_vel:  -76.06  y_vel:   155.94
time_det:   80.20
time_limit:   80.42
Htype: 3  veh_type: -1  x_pos:  -915.10  y_pos: -6781.20  z_pos:     0.00
x_vel:  -60.30  y_vel:   104.44
COVARIANCE
100.000000 0.000000 0.000000 0.000000 0.000000
0.000000 100.000000 0.000000 0.000000 0.000000
0.000000 0.000000 1000000.000000 0.000000 0.000000
0.000000 0.000000 0.000000 100.000000 0.000000
0.000000 0.000000 0.000000 0.000000 100.000000
COVARIANCE
100.000000 0.000000 0.000000 0.000000 0.000000
0.000000 100.000000 0.000000 0.000000 0.000000
0.000000 0.000000 1000000.000000 0.000000 0.000000
0.000000 0.000000 0.000000 100.000000 0.000000
0.000000 0.000000 0.000000 0.000000 100.000000

number of meas: 1
time_det:   82.40
time_limit:   82.62
Htype: 3  veh_type: -1  x_pos: -2097.00  y_pos: -3305.80  z_pos:     0.00
x_vel:  -76.06  y_vel:   155.94
COVARIANCE
100.000000 0.000000 0.000000 0.000000 0.000000
0.000000 100.000000 0.000000 0.000000 0.000000
0.000000 0.000000 1000000.000000 0.000000 0.000000
0.000000 0.000000 0.000000 100.000000 0.000000
0.000000 0.000000 0.000000 0.000000 100.000000
```

Table 3.2-5 Message Buffer from ppm Module

```
meas.num_received: 2
Hmat: 3  x_pos: -2120.500000  y_pos: -3391.399902  z_pos: 0.000000
x_vel: -76.058159  y_vel: 155.940399  time_det: 80.199997  veh_type: -1

meas.num_received: 2
Hmat: 3  x_pos: -915.099976  y_pos: -6781.200195  z_pos: 0.000000
x_vel: -60.300507  y_vel: 104.442368  time_det: 80.199997  veh_type: -1
CV3.pred_state[0]  -2076.038818
CV3.pred_state[2]  -4522.953125
CV3.pred_state[4]  0.000000

CA3.pred_state[0]  9032.459961
CA3.pred_state[3]  -12493.008789
CA3.pred_state[6]  0.000000
IN AVG_PREDICTION: avg_pred_state[0] = -2076.038818
IN AVG_PREDICTION: avg_pred_state[1] = -78.707680
IN AVG_PREDICTION: avg_pred_state[2] = 73.328354
IN AVG_PREDICTION: avg_pred_state[3] = -4522.953125
IN AVG_PREDICTION: avg_pred_state[4] = 110.053093
IN AVG_PREDICTION: avg_pred_state[5] = -52.985432
IN AVG_PREDICTION: avg_pred_state[6] = 0.000000
IN AVG_PREDICTION: avg_pred_state[7] = 0.000000
IN AVG_PREDICTION: avg_pred_state[8] = 0.000000


num_target: 1  num_meas: 2
After LOG_LIKE
3.019633e+01  5.220504e+01

assign[0]: 0

MODEL_PROBABILITY: model_probability[0] = 1.000000
 CV3.likeihood_val = 1.000000e+00
MODEL_PROBABILITY: model_probability[1] = 0.000000
 CA3.likeihood_val = 0.000000e+00


FUSION STATE(SEND to DISPLAY)
       X = -2119.750488
       Y = -3399.716309
       Z = 0.000000

------------TRACK TRAVERSING------------
track_id: 1
veh_type: 102
associated_meas: 0
FUSION X= -2119.75  Y= -3399.72  Z=     0.00
```

Table 3.2-6 Track Information from Tracker Module

```
meas.num_received: 1
Hmat: 3  x_pos: -2097.000000  y_pos: -3305.800049  z_pos: 0.000000
x_vel: -76.058159  y_vel: 155.940399  time_det: 82.400002  veh_type: -1
CV3.pred_state[0]   -2301.229248
CV3.pred_state[2]   -2904.831787
CV3.pred_state[4]   0.000000

CA3.pred_state[0]   -2214.611084
CA3.pred_state[3]   -2958.902832
CA3.pred_state[6]   0.000000
IN AVG_PREDICTION: avg_pred_state[0] = -2301.229248
IN AVG_PREDICTION: avg_pred_state[1] = -82.490204
IN AVG_PREDICTION: avg_pred_state[2] = 0.000000
IN AVG_PREDICTION: avg_pred_state[3] = -2904.831787
IN AVG_PREDICTION: avg_pred_state[4] = 224.946884
IN AVG_PREDICTION: avg_pred_state[5] = 0.000000
IN AVG_PREDICTION: avg_pred_state[6] = 0.000000
IN AVG_PREDICTION: avg_pred_state[7] = 0.000000
IN AVG_PREDICTION: avg_pred_state[8] = 0.000000


CV3.pred_state[0]   -1047.761353
CV3.pred_state[2]   -6551.426758
CV3.pred_state[4]   0.000000

CA3.pred_state[0]   -1047.761475
CA3.pred_state[3]   -6551.426758
CA3.pred_state[6]   0.000000
IN AVG_PREDICTION: avg_pred_state[0] = -1047.761475
IN AVG_PREDICTION: avg_pred_state[1] = -60.300507
IN AVG_PREDICTION: avg_pred_state[2] = 0.000003
IN AVG_PREDICTION: avg_pred_state[3] = -6551.426758
IN AVG_PREDICTION: avg_pred_state[4] = 104.442284
IN AVG_PREDICTION: avg_pred_state[5] = -0.000005
IN AVG_PREDICTION: avg_pred_state[6] = 0.000000
IN AVG_PREDICTION: avg_pred_state[7] = 0.000000
IN AVG_PREDICTION: avg_pred_state[8] = 0.000000
```

Table 3.2-7 Track Information from Tracker Module

```
num_target: 2  num_meas: 1
After LOG_LIKE
5.464704e+02
3.186033e+03

assign[0]: 0
assign[1]: -1

MODEL_PROBABILITY: model_probability[0] = 0.000000
 CV3.likeihood_val = 6.250718e-35
MODEL_PROBABILITY: model_probability[1] = 1.000000
 CA3.likeihood_val = 9.657914e-01


FUSION STATE(SEND to DISPLAY)
       X = -2141.317627
       Y = -3205.733398
       Z = 0.000000

FUSION STATE(SEND to DISPLAY)
       X = -1047.761475
       Y = -6551.426758
       Z = 0.000000

------------TRACK TRAVERSING------------
track_id: 1
veh_type: 102
associated_meas: 0
FUSION X= -2141.32  Y= -3205.73  Z=     0.00

------------TRACK TRAVERSING------------
track_id: 2
veh_type: -1
associated_meas: -2
FUSION X= -1047.76  Y= -6551.43  Z=     0.00
```

Table 3.2-7(cont'd)   Track Information from Tracker Module

The same file iew_d2e2 has been used to generate figures 3.2-1 through 3.2-6. Simulated trajectories of two A-7's, vehicle numbers 21 and 22, are included in the file. The same models cv3 and ca3 with the same parameters are utilized.

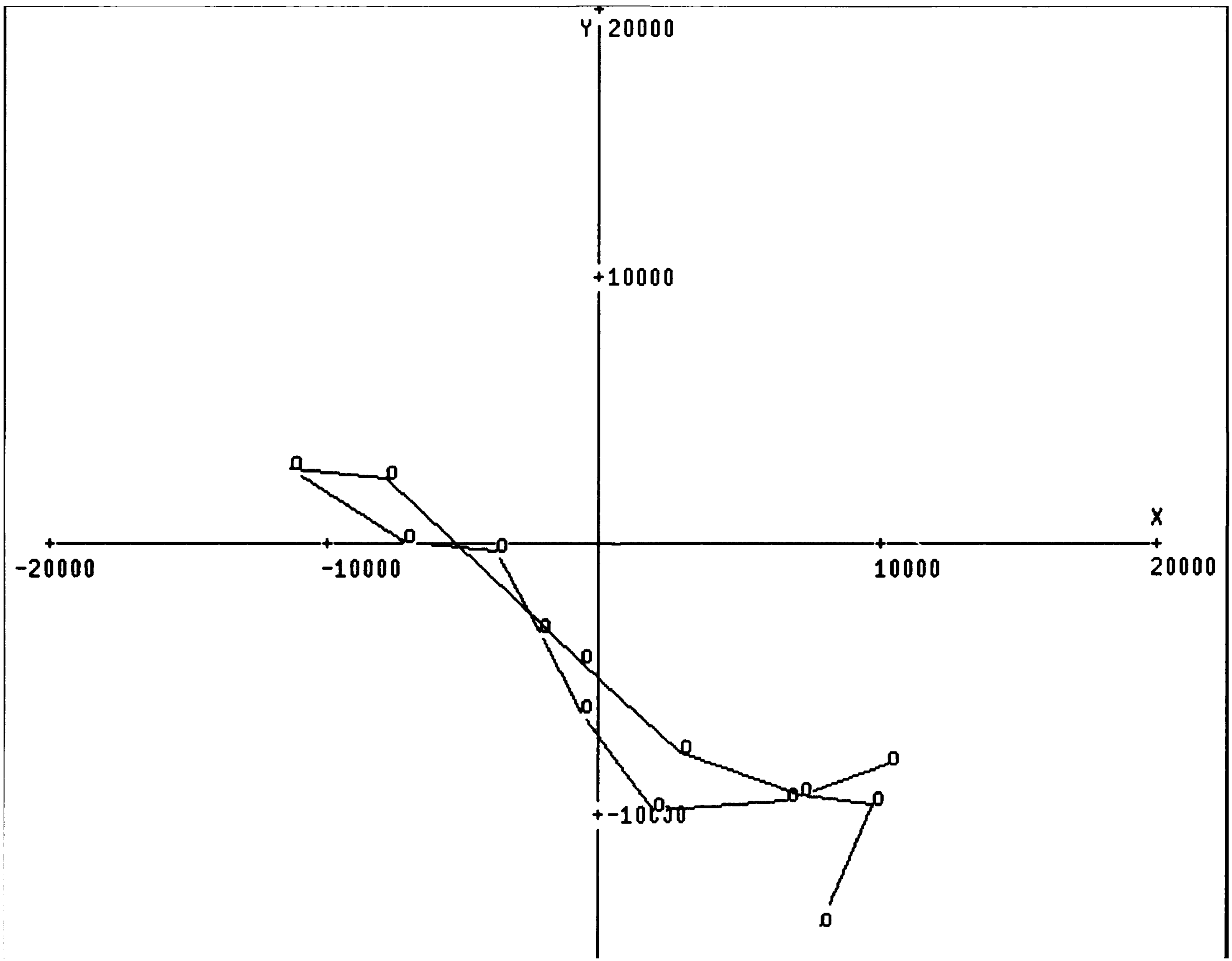
Figures 3.2-1 and 3.2-4 are simulated trajectories without measurement noises for vehicle 21 and vehicle 22, respectively. Since the MMAS file provides measurements without noise when the corresponding target is not detected, these are obtained by extracting only undetected measurements for each vehicle. It means that each of figures does not show the whole trajectory since the positions of targets are missed in these figures. However, they provide the trends of movement of the targets in general.

From the MMAS file, the detected measurements for vehicle 21 were obtained, and utilized as an input to the CIP tracking system. In this test, gating was excluded in data association and blocking netGETANY was used in tracker module instead of nonblocking netGETANY_NBLK.

Figure 3.2-2 shows tracking results. In this test, since only one measurement was used at a time and gating was not included, the Munkres's algorithm gives automatic association, i.e., one measurement for one track. Hence, this result shows the performance of the SRIF with the IMM.

Figure 3.2-5 also shows the trajectory for the vehicle 22 obtained under the same conditions. As can be seen in these figures, the behaviors of the trajectories are almost same as the trajectories in figures 3.2-1 and 3.2-1 except starting points and ending points. These differences are due to the fact that tracking starts only when the first measurements are received by the tracker and that figure 3.2-1 and 3.2-2 contain information about undetected measurements as soon as the simulation event starts.

Figure 3.2-3 and 3.2-6 show tracking results when multi-target measurements are utilized. In these cases, difficulty in data association has been observed. We believe that this is due to mainly sampling rate. The file utilized here, iew_d2e2, has reporting step 20 seconds.


Y
20000
+10000
-10000
X
-20000
-10000
10000
20000

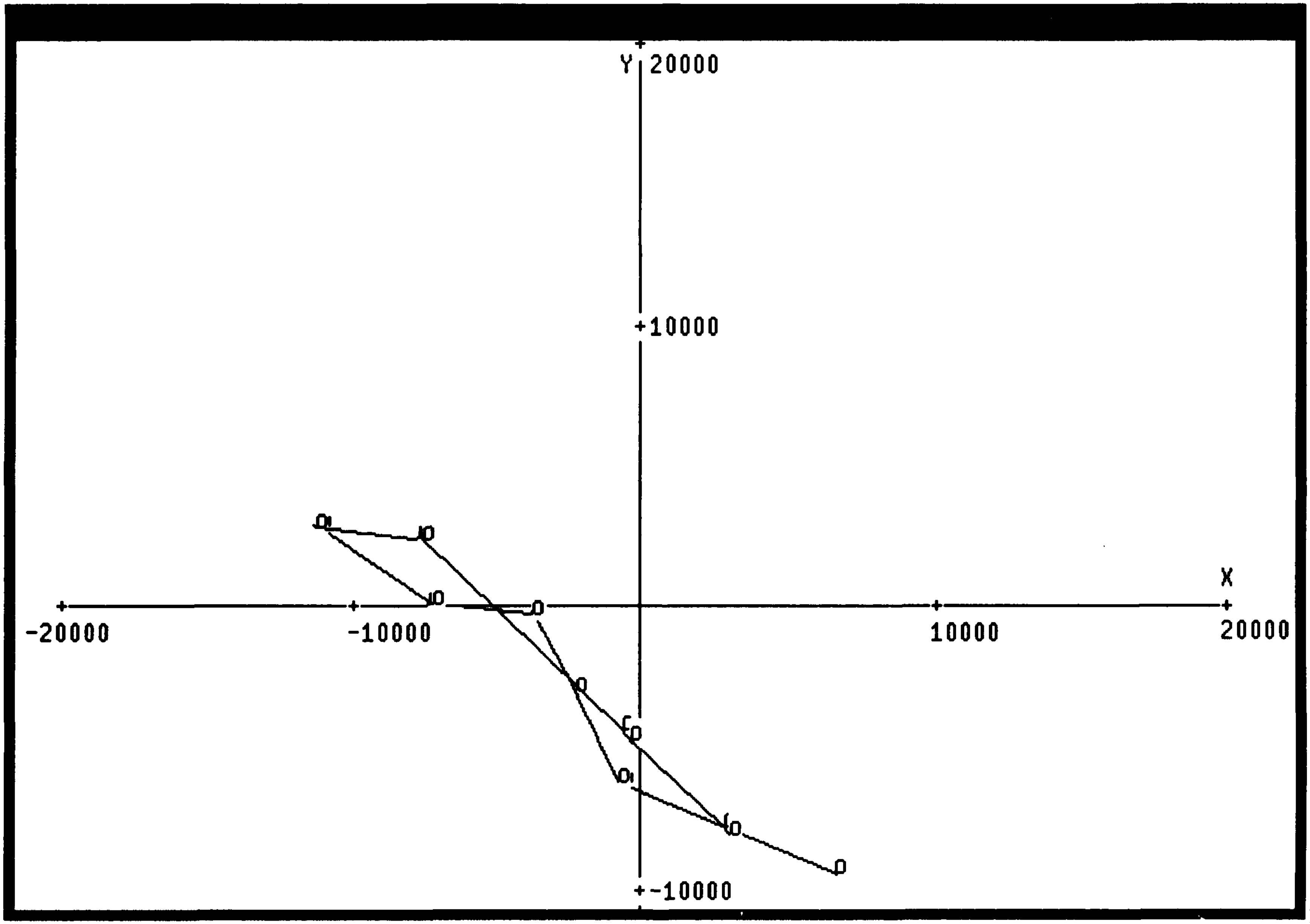


Figure 3.2-2 Tracking Results with A Single Target Measurements

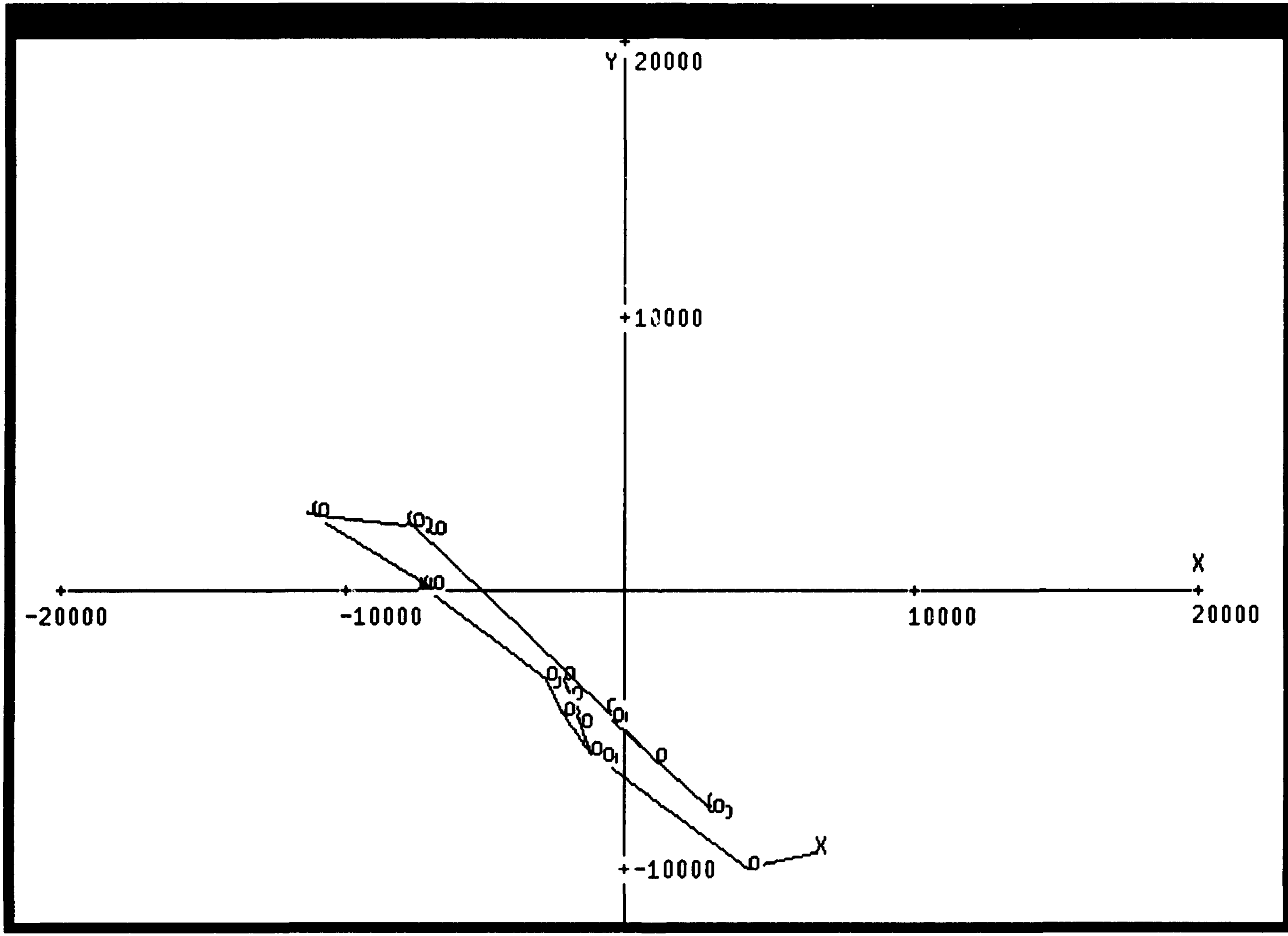


Figure 3.2-3 Tracking Results with Multi-Target Measurements

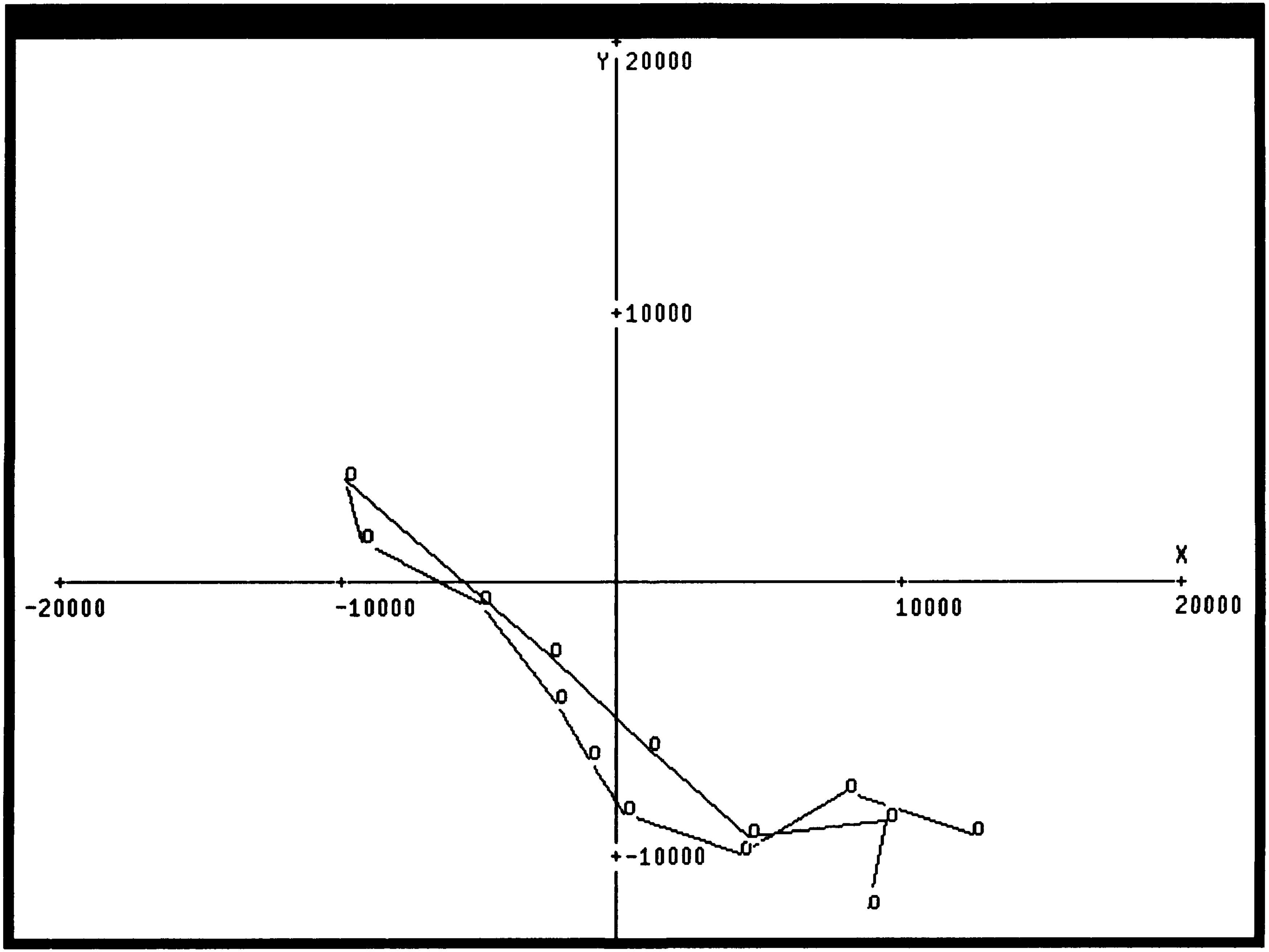


Figure 3.2-4 True Trajectory of Vehicle 22

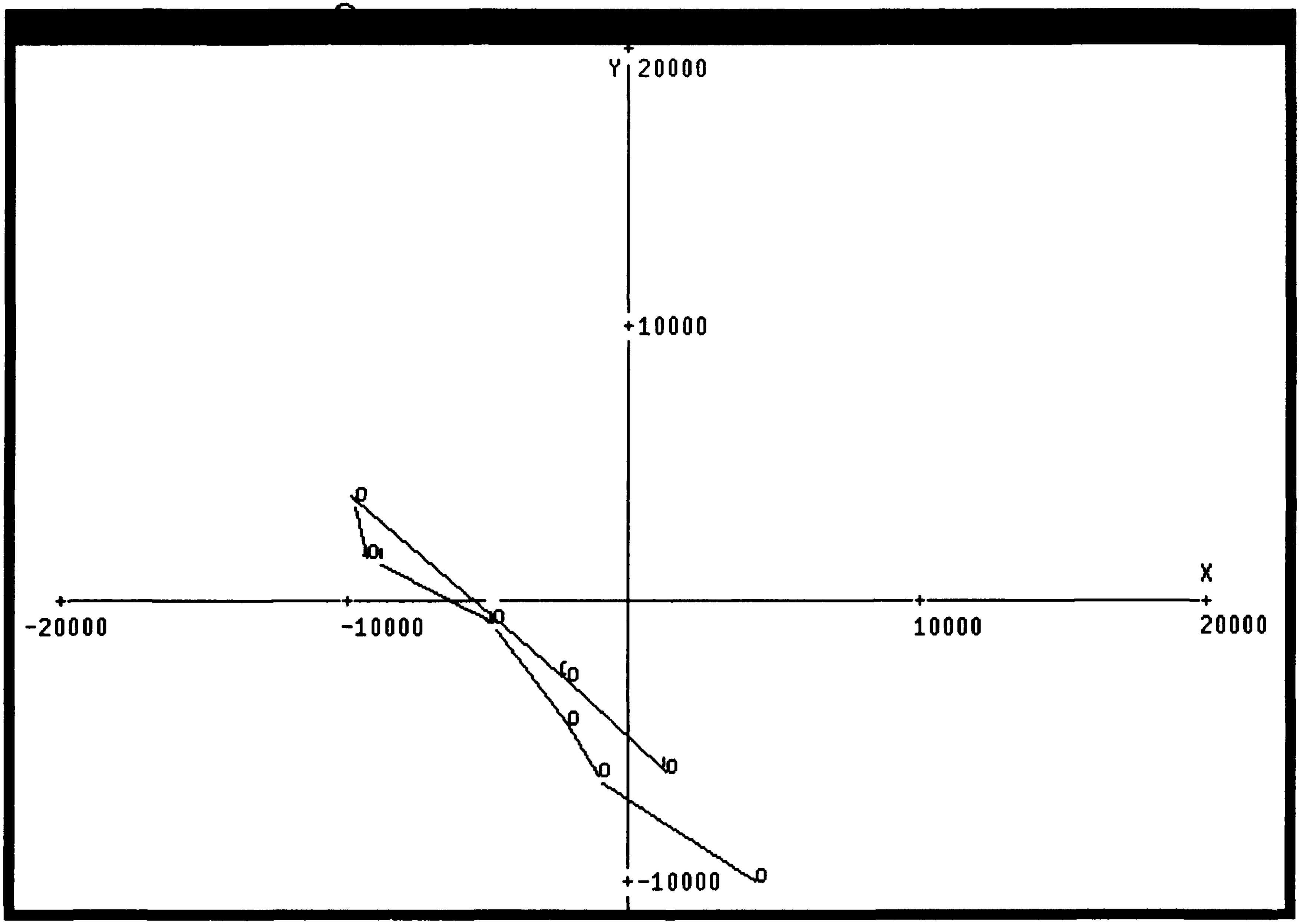


Figure 3.2-5 Tracking Results with A Single Target Measurements

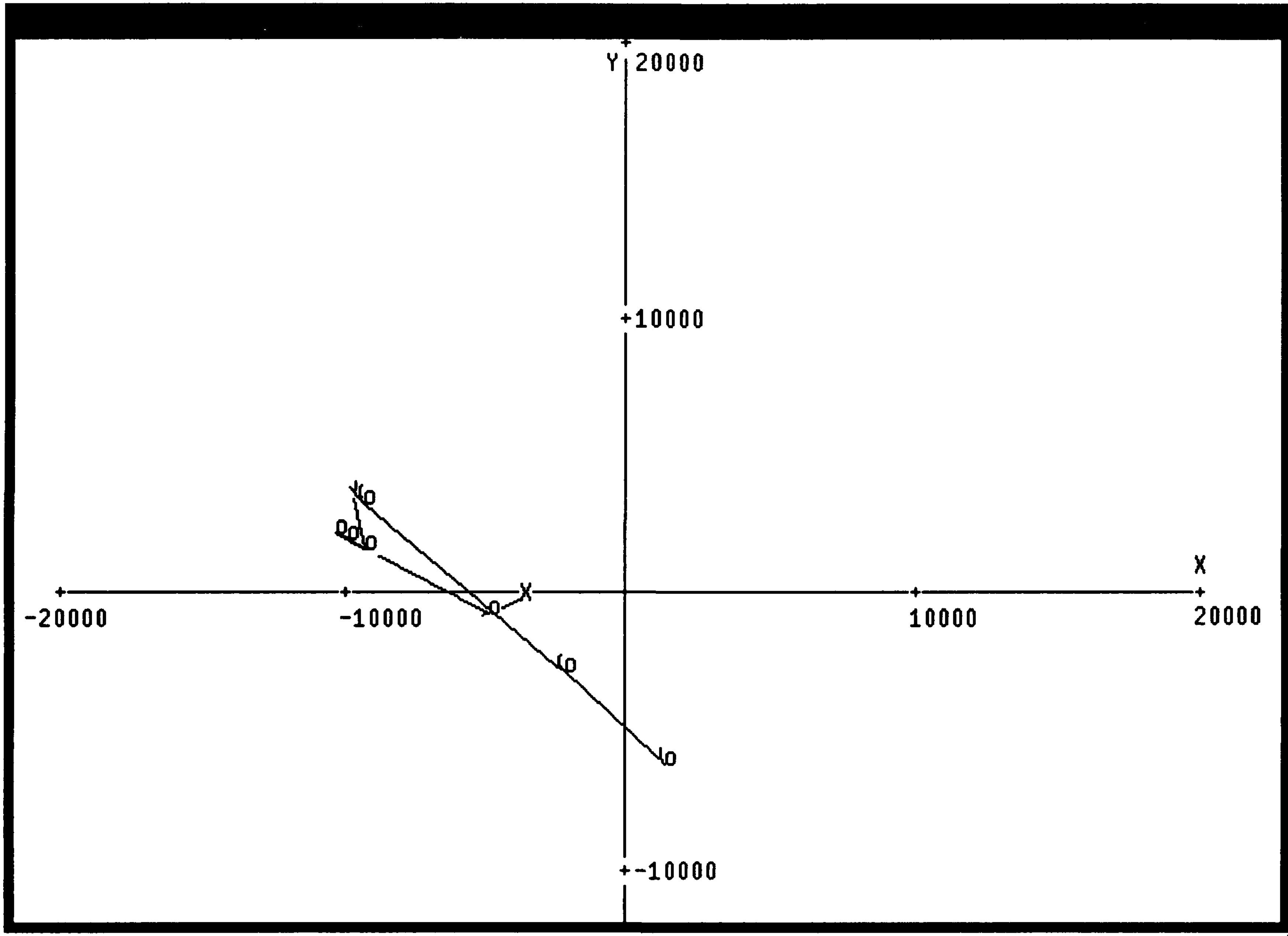


Figure 3.2-6 Tracking Results with Multi-Target Measurements

4. Concluding Remarks and Recommendations

Based on the test of the HDL/CIP tracking system on the MMAS simulation data file, iew_d2e2, the conclusions are summarized as follows.

The test shows that the performance of the SRIF with the IMM is in an acceptable range. That is, even though the targets are highly maneuvering, the averaged filtered state from selected models closely follows the true trajectory without measurement noise. However, the disadvantage of this approach is in the computational load due to the use of multiple filters.

It has been observed that the choices of design parameters affect the tracking performance. For example, if the parameter DATA_ACCUMULATION_PERIOD is decreased, the chances of fusion are also decreased since the number of measurements inside of time alignment period becomes smaller. On the other hand, if the same parameter is increased, the chances of fusion are also increased. This means that the set of measurements received by tracker module will be different whenever different value is assigned. Hence, different tracking results are expected sometimes.

The smaller the DATA_ACCUMULATION_PERIOD is, the more measurements are regarded as distinct. The smaller value also will stabilize the tracking performance. However, the advantages of fusion will be lost. Determination of an optimal value, which distinguishes enough of the measurements, but does not lose the advantage of fusion is a difficult task, since it depends on the situation. More tests on real data might give some clue about this point.

Also, other parameters, such as error covariance matrices, generate effects which are difficult to estimate. In the SRIF formulation, information matrices are utilized which are derived from covariances. If large numbers are assigned as entries of covariances, it will generate small numbers in the information matrix, and may result in numerical instability.

Detailed study is required to get a reasonable bound for these numbers.

As seen in section 3.2, the true trajectories of targets included in the MMAS files show very similar behaviors. Soft decision scheme was employed in the tracker software, which is basically based only on kinematic information. These two points make data association hard. To get more accurate data association results, not only kinematic data but other information, possibly including nonnumeric attributes, should be integrated.

The following issues are recommended for the further enhancements of the HDL/CIP tracking system.

The predicted state estimate is dependent on models utilized in filtering. Also, the movement of some vehicles, especially ground vehicles, is dependent on terrain. This makes the reliability of the predicted state of ground vehicles low. An expert system which can combine the kinematic information from filtering and other information such as terrain information should be considered.

Vehicle type information is utilized in the tracking process. The procedure employed to determine vehicle type is very simple, and utilizes kinematic information only. The misclassification of vehicle type also leads to unexpected tracking results. As in recommendation 1), development of a system which can integrate numerical and nonnumerical information to get a more reliable vehicle type is required.

An approach such as the maximum likelihood method should be considered to estimate reasonable values of parameters. The parameter estimation procedure and tracking procedure should be performed concurrently. Parameters obtained should be implemented in the tracking system by user's request.

For the measurements such as the MMAS data, the MHT method is more suitable than any method employing a soft decision scheme. In the implementation of the MHT, hypotheses generated must be pruned. The development of the criteria to prune hypotheses should be considered. A hybrid system , which can integrate kinematic information and the results from the data association method employed in tracker software, and some nonnumeric information together will give more reliable data association results.

Finally, a more user friendly interface should be developed.

References

[1] G. Bierman, "Factorization Methods for Discrete Sequential Estimation," Academic Press, 1977.

[2] Y. Bar-Shalom, T. Fortman, "Tracking and Data Association," Academic Press, 1988.

[3] M. Belzer, G. Bierman, "Maximum Likelihood Estimation Using the Square Root Information Filters," pp. 1293-1298, IEEE Transaction on Automatic Control, vol 35, no 12, Dec. 1990.

[4] F. Burgeois, J-C Lassalle, "An Extension of the Munkres Algorithm for the Assignment Problem to Rectangular Matrices," pp. 802-806, Communications of the ACM, vol 14, Dec. 1971.

[5] S. Blackman, "Multiple-Target Tracking with Radar Application," Artech Kouse, 1986.

[6] C. Chang, L. Youens, "Measurement Correlation for Multiple Sensor Tracking in a Dense Target Environment," pp. 1250-1252, IEEE Transactions on Automatic Control, vol AC-27, no 6, Dec. 1982.

[7] A. Houles, Y. Bar-Shalom, "Multisensor Tracking of a Maneuvering Target in Clutter," pp. 176-189, IEEE Transaction on Aerospace and Electronic Systems, vol AES-25, no 2, March, 1989.

Distribution List

US Army LABCOM
Adelphi Laboratory Center
2800 Powder Mill Road
Adelphi, Maryland 20783-1197
ATTN: Shipping & Receiving, Building 102
M/F: Mr. Larry Tokarcik/SLCHD-TA-AS

US Army LABCOM
Adelphi Laboratory Center
2800 Powder Mill Road
Adelphi, Maryland 20783-1197
ATTN: Mrs. Pauline Baumgartner/SLCHD-TS-S

DTIC-FDAB
Defense Technical Information Center
Cameron Station
Alexandria, VA 22304-6145